Z MIKROKOMPUTEREM NA TY

5

Bajtek

NR INDEKSU 353965
PL ISSN 0860 – 1674

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH

NR 5(29) MAJ 1988 CENA 150ZŁ

NA POCZĄTKU BYŁA
LICZBA
KONKURS MATEMATYCZNY

KATALOG DYSKOWY

ZUPA
Z GWOŹDZIA

SM SZTANDAR MŁODYCH

KOMPUTER DLA HUMANISTY

Komputery karmione liczbami trawiły je, dodawały, odejmowały i produkowały następne liczby. Nie wiadomo dziś dokładniej kiedy i gdzie pojawiło się na ekranie pierwsze słowo. Za sprawą komputerów i DTP przyszłość naszej cywilizacji może wyglądać zgoła inaczej niż sądzili do niedawna futurolodzy.

REWOLUCJA PRZY FLEET STREET

STARQUAKE

PLANY I KLANY

Przed miesiącem mogliście na tym miejscu przeczytać informację o „bajtkowym” spotkaniu w Jachrance, podczas którego dyskutowaliśmy m.in. o tym, co chcielibyśmy zmienić w sposobie redagowania naszego miesięcznika, a już dziś sięgając po kolejnego „Bajtka” znajdziecie w nim kilka naszych nowych propozycji. Postanowiliśmy, m. in. pokazać Wam, że i my, „ośmiobitowcy” nie musimy być tak bardzo w tyle za użytkownikami maszyn typu PC i śmiało używać możemy naszych „zabawek” do całkiem poważnych zastosowań. Chcielibyśmy, aby szkolne, czy osiedlowe kluby komputerowe znalazły sobie przy okazji nowe pola do działania – skomputeryzowane biblioteki, podjęcie się wydawania gazetki, itp. Komputery domowe, problemy z ich upowszechnianiem, programowaniem mogą całkiem dobrze przybliżyć Wam problemy użytkowników profesjonalnych.

Z tych właśnie powodów począwszy od niniejszego numeru będziemy starali się o to, by każdy „Bajtek” miał swoją „cover story”, czy inaczej motyw przewodni obecny w nim od okładki poprzez część publicystyczną, aż do klanów. Nie oznacza to oczywiście, że zrezygnujemy z opisów gier, czy tak popularnych cykli, jak „następny krok”. Doszliśmy jednak do wniosku, że, być może, w naszym wykładzie o komputerach przydałoby się więcej systematyki.

Jak widzicie zrezygnowaliśmy również z zamieszczania spisu treści aktualnego numeru na rzecz zapowiedzi kolejnego. Myślimy, że nasi stali czytelnicy i tak bez trudu dotrą do „swoich” klanów, czy opisów gier. Układ numeru pozostawiliśmy bowiem bez zmian.

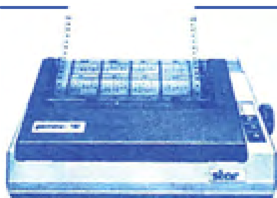
Nasz upór w pokazywaniu uroków ośmiobitowców może wydawać się nieco irracjonalny. Przecież, jak sami o tym piszemy na str. 20–21, w reportażu z tegorocznych targów CeBIT, konstrukcje te powoli znikają z ofert światowych potentatów. Jednak jednocześnie nie widać, aby sprzęt 16-bitowy staniał na tyle dramatycznie, by było nań stać przeciętnego polskiego inżyniera, nie mówiąc już o polskim Świętym Mikołaju. 520 ST pod rodzimą choinką, to jednak wciąż bajka.

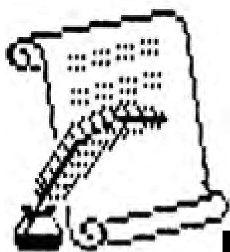
Naszego przywiązania do sprzętu domowego nie podzielają niestety firmy komputerowe działające w Polsce. Zapatrzone w milionowe zamówienia i takież zyski ze sprzedaży IBM i podobnych maszyn pozostawiły na lodzie kluby i kółka komputerowe, a także tych prywatnych użytkowników, którzy chcieliby np. naprawić swój Commodore, czy Amstrad. Jeśli nie liczyć dosłownie kilku takich placówek jak sklep „Bajtka” w Bytomiu, czy CSH (tam już też przeważają IBM-y) komputer czy urządzenie peryferyjne za złotówki kupić można tylko na giełdzie.

Na szczęście (?) nie pozostaliśmy jedynymi epigonami. W Ministerstwie Edukacji ciągle wprawdzie trwają dyskusje na temat „Junior” czy coś innego, ale z pewnością wybór padnie na ośmiobitowca. Może zatem nie musimy się bać o to, że z konieczności staniemy się konkurencją dla „Mikroklanu” czy „Komputera” na rynku czasopism o PC.

Na koniec jeszcze słów kilka o „bajtkowych” zamierzeniach. Już wkrótce znajdzie się wydany przez nas kolejny „user”, specjalny numer „Tylko o Commodore”. Wszystkim prenumeratorom przypominamy: będziecie musieli niestety powalczyć o niego w kioskach. Do kiosków zaglądać powinni także i fani „Atari”. Wkrótce znajdziecie tam „Tylko o Atari-2”. Podobają wam się nasze pomysły? Jeśli nie – napiszcie, jeśli tak – napiszcie również.

Grzegorz Onichimowski





GAZETOWY KONKURS BAJTKA"!

Numer, który macie w ręku poświęciliśmy słowom i ich przetwarzaniu przez mikrokomputer. Być może uda nam się przekonać Was do tego, że przy pomocy amatorskiego sprzętu i niezbyt wyszukanych programów stać Was na uruchomienie drukami we własnym domu, w szkole, czy w klubie osiedlowym.

Zapraszamy zatem wszystkich do współzawodnictwa. Do końca września br. nasza redakcja oczekiwać będzie na wydane techniką komputerową gazety. Ich edytorami mogą być szkolne lub osiedlowe kluby mikrokomputerowe. Mogą to być także gazety rodzinne.

Oceniać będziemy wszystko — szatę graficzną, pomysły softwarowe wykorzystane do stworzenia Waszej gazety, ale także i jej treść. Pamiętajcie, że sama forma, choćby najładniejsza, nie zapewnia poczytności tytułu prasowego. Sprawdźcie się zatem jako komputerowcy i jako dziennikarze.

Na zwycięzców konkursu gazetowego czekają cenne nagrody. Pierwszą będzie ufundowana przez firmę „Star” i jej dystrybutora na Polskę — „ABC Data” profesjonalna drukarka NX-15. Jest zatem o co walczyć.

Życzymy powodzenia i przypominamy: Termin nadsyłania gazetek (prosimy po trzy egzemplarze) upływa 30 września br.

ZA MIESIĄC

● ● ● ● ● ● ● ●

W następnym numerze "Bajtka" przeczytacie m. in:

- Czy musimy być rajem dla piratów, a piekłem dla informatyków, czyli sonda "Bajtka" na temat niezbędnych uregulowań prawnych dotyczących polskiego rynku softwarowego
- antypirackie porady praktyczne, czyli czy chronić programy i jak to robić
- Atari 65XE Game
- Watson, polski przyjaciel Sherlocka
- Chimera

Z TEKI CHOCHLIKA

Autorem fotoreportażu z imprezy Komputer'88 (Bajtek 3/88) jest Krzysztof Wojciewski. Przepraszamy za pominięcie jego nazwiska.

Ze względu na długi cykl wydawniczy naszego pisma zamieściliśmy w nim, niestety, nieaktualne ceny na wyroby sprzedawane przez firmę "Polanglia". Nowe (niższe!) mogę czytelnicy uzyskać dzwoniąc do firmy. Tel. London 840 1715.

W numerze BAJTKA 4/88 zaszła pomyłka w opisie rysunków w KLANIE COMMODORE w artykule ELEKTRONIKA JOYSTICKA. Rysunek oznaczony jako 1 odnosi się do Commodore 128 natomiast rysunek 2 do Commodore 64, a nie odwrotnie jak sugeruję opisy. Za pomyłkę serdecznie Czytelników przepraszamy.

„BAJTEK” — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH” ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21 -12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański- -redaktor naczelny „Sztandaru Młodych”.

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM” — kierownik zespołu „Bajtka”), Roman Poznański (z-ca sekretarza redakcji „SM” — sekretarz zespołu „Bajtka”), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Gogolewski, Andrzej Kowalewski, Andrzej Podulka, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski.

Klany redagują: Commodore — Klaudiusz Dybowski, Amstrad-Schneider — Tomasz Pyć, Sergiusz Wolicki, Spectrum — Marcin Przasnyski, Michał Szuniewicz, Atari — Wiesław Migut, Wojciech Zientara.

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wółtańska

WYDAWCA: RSW „Prasa-Książka-Ruch” Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414. Cena 100 zł.

Skład techniką CRT-200, przygotowalnią offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA- -KSIĄŻKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51. Nr zlecenia 010188 n. 150.000 egz. U-113



Bajtek

PIĘTNO OSOBOWOŚCI

Rozmowa z inż. Jackiem Sobczykiem, współwłaścicielem firmy „Cyfronex".



— Wydawanie gazety nie ruszając się zza biurka to na świecie już prawie standard i do tego — jak sądzę — nie muszę pana przekonywać.

— Zgadza się.

— Jak pan zatem przekonuje potencjalnych klientów, że warto zainwestować w oferowaną przez waszą firmę polską wersję „Desktop publishing"?

— W Polsce mało jest przykładów komputerowego profesjonalnego składu i łamania gazet i czasopism. Z różnych przyczyn zatrzymał się ten proces na etapie składu i korekty. „Przegląd Tygodniowy", który stara się z naszą pomocą rozwiązać rzecz kompleksowo, jest w gruncie rzeczy jeszcze w fazie eksperymentu — fakt, że zaawansowanego. Żeby zatem przekonać potencjalnego klienta o zaletach naszego systemu, zapraszam go do firmy, żeby pokazać mu, czym dysponujemy, co może osiągnąć.

— Zwykle program demonstracyjny obliczony jest na zrobienie dobrego wrażenia!

— Toteż taka demonstracja możliwości użytkowych jest zwykle wstępem do konkretnej rozmowy. Staramy się poznać problemy klienta, dowiedzieć się w jakich warunkach pracuje, co drukuje, w jakich ilościach, jak wygląda jego dotychczasowy warsztat pracy. Mając te i inne wiadomości staramy się urządzić pokaz, który odpowiedziałby nie werbalnie, lecz faktycznie na jego pytania i wątpliwości.

— Jakim minimalnym zestawem sprzętu musi dysponować przyszły użytkownik?

— Wydawanie gazety zza biurka obejmuje dużo więcej niż sam proces druku. Zelektronizowana redakcja to elektroniczna biblioteka wycinków, automatyczne zbiera nie danych z dalekopisów etc, etc. My zapewniamy automatyzację tylko samego procesu wydawniczego. Minimum sprzętu, który jest przez nas wymagany to mikrokomputer zgodny z IBM PC/XT z kartą Hercules, odpowiednim monitorem i 2 napędy dysków elastycznych.

Oczywiście, kiedy będzie to komputer AT i twardy dysk polepszy się jakość i zwiększą się możliwości, zwłaszcza podczas pracy w podglądzie graficznym.

— Wasza firma nie prowadzi, wzorem innych z tej branży, agresywnego marketingu. Prawdę mówiąc ofert „Cyfronexu" w prasie nie spotyka się w ogóle.

— Z jednej strony ma pan rację, z drugiej nie. Zaczniemy od drugiej — otóż naszym sztandarowym wyrobem są komputery poligraficzne POLTYPE, umożliwiające profesjonalny skład dowolnych tekstów. Ich głównymi odbiorcami są obecnie drukarnie, niektóre wydawnictwa, placówki naukowe, prywatne firmy poligraficzne. Do tego grona docieramy poprzez udział w branżowych targach, fachowych pokazach i seminariach. Ma pan natomiast rację, że bardziej agresywny marketing jest niezbędny do wyjścia poza tradycyjny krąg klientów. W firmie jesteśmy zgodni co do tego — różnimy się w opiniach co do momentu rozpoczęcia kampanii reklamowej. Mamy bowiem przekonanie, że istniejące systemy mogą rozwiązać problemy wydawnicze wielu instytucji spoza zawodowej poligrafii, lecz jednocześnie wiemy, że niedługo będziemy mogli zaoferować system łączący profesjonalizm dotychczasowych z łatwością obsługi typowego „desktop publishing".

— I mając taki sprzęt plus program można każdą ze stron „Bajtka" złożyć, przeprowadzić korektę i złamać — czyli ułożyć na kolumnie tak jak będzie wyglądała po wydrukowaniu. Co dalej?

— Dyskietki zawierające ostateczną, bezbłędną wersję publikacji należy przekazać drukarni do naświetlenia. Większość drukarni w Polsce wyposażona jest w naświetlarki, z których gros to produkty firmy Monotype, z którą współpracuje nasz program.

— Czyli nie wystarczy komputer i program — bez drukarni ani rusz!

— Jeżeli chce się wydawać profesjonalną gazetę, to tak. W przypadku publikacji biuletynów np. wystarczy w zupełności jakość wydruku z drukarki laserowej.

— Ostateczną, bezbłędną wersję publikacji zapewnia tzw. podgląd graficzny, który pozwala obejrzeć na ekranie monitora obraz graficzny składanej publikacji. Co to takiego?

— Jest to całkowicie nowa funkcja w mikrokomputerach POLTYPE-03/04. Ekran monitora jest tu jak gdyby oknem, które można przesuwać w kierunku pionowym i poziomym w celu obejrzenia całej strony składu. Podgląd graficzny umożliwia wykrycie i usunięcie wszystkich błędów składu.

— Kto jest autorem programów w waszej firmie?

— Głównym autorem jest jeden ze współwłaścicieli — inż. Krystian Rudo.

— Nie życzę źle mózgowi firmy — ale może on na przykład zachorować lub wpaść pod samochód.

— No cóż, również źle nie życzę panu Rudo i wolałbym nie rozpatrywać takich okoliczności. Gdyby jednak zaistniał któryś z tych faktów, to prace uległyby zahamowaniu na pewien czas, mam nadzieję, że krótki, i dalej będą się rozwijały z piętnem innej osobowości.

— Jak w wielu innych dziedzinach tak i w DTP obowiązują standardy, pewne normy, którym przewodzą słynne już Page Maker i Ventura. Jak do tych, sprawdzonych osiągnięć światowych, ma się produkt, który oferuje firma Cyfronex?

— Podchodzimy jak gdyby z drugiej strony. Zarówno Ventura jak i Page Ma-

ker są programami, które, wywodząc się z automatyzacji prac biurowych, wypracowały pewne standardy obsługi, klasyczny wygląd ekranu i sposoby komunikacji z użytkownikiem.

— Łatwe i wygodne.

— Przyznaję, że dla nieprofesjonalistów tak. Natomiast, jeśli chodzi o zawodową poligrafię, programy te zawierają istotne ograniczenia — nie zawsze potrafią zrobić wszystko z punktu widzenia typografii. My rozpoczęliśmy działalność z drugiego jakby końca — od współpracy z drukarniami, realizując na ich zlecenia programy spełniające najbardziej fantazyjne możliwości, które można uzyskać na ich sprzęcie! Idąc tą drogą, od profesjonalnego jądra, staramy się dorobić interfejs użytkownika symulujący standardy, które się przyjęły.

— Jednak w porównywaniu do standardów wasz interfejs jest graficznie dość ubogi. Brak okien, piktogramów, menu, itd., które tam prowadzą laika „za rękę". Trudno skorzystać z waszego systemu, bez podstawowego kursu poligrafii.

— Potwierdza pan w ten sposób jego profesjonalne zalety. Program i system tworzony był początkowo na zamówienie drukarni, a więc profesjonalistów absolutnych. Następnie modyfikowany dla redakcji, wydawnictw i działów małej poligrafii, gdzie pracują być może nie zawodowcy w tej dziedzinie, ale fachowcy mający z nią kontakt na codzień. Dlatego im łatwiej „chwycić" posługiwanie się naszym programem nie mającym jeszcze pełnego interfejsu graficznego niż przeciętnemu użytkownikowi. Takiemu na przykład, który eksploatuje system DTP na Zachodzie, nie mając praktycznie do czynienia z drukarnią, nie chce mieć z nią do czynienia i nie będzie miał, kreując publikację w oparciu o modelowy zestaw hardwarowy (IBM PC, scanner, drukarka laserowa) plus program faktycznie „na stole".

— Ale przecież, firma przymierza się zarówno do „okien" jak i piktogramów, menu etc.

— Oczywiście, z tym że nie chcemy bezkrytycznie małpować tego, co już zostało zrobione, ponieważ nie uważamy, że wszystkie rozwiązania są najlepsze. Ja, na przykład, nie będąc fachowcem w dziedzinie poligrafii, muszę do czasu do czasu coś przygotować do druku i wtedy nie lubię przedzierać się przez stosy piktogramów, menu i innych świecących funkcji na ekranie.

— Nie zawsze to co pan lubi doceniają klienci — a firma żyje z tego, co oni kupią. Poza tym co innego małpowanie, a jeszcze coś innego dopasowanie się do standardów i norm.

— Bezwzględnie potrzebne jest dopasowanie się na pewnym etapie nie do znanych programów DTP, a raczej wszystkich innych, które z nimi współpracują: baz danych, elektronicznych arkuszy obliczeniowych i nawet typowych edytorów — żeby zapewnić przepływ danych między tamtymi a naszym programem. Ważna jest współpraca w standardowym otoczeniu, a nie naśladowanie typowych ikon i menu. Naszą filozofią jest dodawanie funkcji, która potrafi w sposób logiczny rozszerzyć krąg naszych użytkowników

— Niewiele polskich firm zajmuje się tą dziedziną. Jakie bariery powstrzymują od inwestycji w DTP?

— Będę brutalny — jest to taka dziedzina, gdzie programy muszą odpowiadać polskim warunkom. Trudno jest uzyskać dobry program przez prostą polonizację programu zachodniego. Trzeba zainwestować bardzo dużo myśli — to z jednej strony, Z drugiej: Rank Xerox jako właściciel Ventury sam chce z nim wejść na polski rynek i dlatego będzie patrzył na ręce potencjalnym piratom.

Rozmawiał:
Franciszek Penczek

NA POCZĄTKU BYŁA LICZBA

Na początku komputerowego świata była LICZBA. Liczby wypełniały ekrany monitorów, długie ich kolumny wystukiwały pracowicie drukarki. Tasiemcowe wydruki zrozumieć mogli tylko wtajemniczeni. Komputery karmione liczbami trawiły je. dodawały i odejmowały i produkowały następne liczby. Nie wiadomo dziś dokładnie, kiedy i gdzie pojawiło się na ekranie pierwsze SŁOWO.

Kiedy jednak już zostało użyte, wiadomo było, że nie ma odwrotu — komputer musiał się nauczyć języka ludzi, bo język słów rozumie więcej ludzi niż język liczb. I chociaż słowa są dla komputera trochę bardziej skomplikowane niż liczby, nie była to przeszkoda poważna. Wszystko co komputery liczą, liczą i tak po swojemu, tylko rezultaty obliczeń muszą na koniec przedstawić zamieniając niektóre liczby na litery, a te połączyć w słowa. Moglibyście się wprawdzie umówić, że np. 528 69 81 oznacza „hej to ja" ale któż by to spamiętał, zwłaszcza, że w systemie dwójkowym, którym posługuje się komputer te liczby byłyby jeszcze dłuższe i jeszcze trudniejsze do zapamiętania.

Kiedy już okazało się ku zaskoczeniu laików, że komputer potrafi pisać po ludzku, wielu z nich zaświtała myśl, by oprócz „zgodnego z ich naturą" liczenia, użyć komputerów do manipulowania słowami. Wówczas to w licznych laboratoriach naukowych zaczęto męczyć „mózgi elektronowe" — jak romantycznie nazywano wtedy komputery — układaniem wierszy. Wiele tych próbek wydrukowały gazety i szczerze mówiąc, niektóre z nich były bardziej zrozumiałe niż utwory współczesnych im awangardowych poetów. Mimo szalonego rozwoju mocy obliczeniowej komputerów ta dziedzina twórczości nie poddaje się (na szczęście) automatyzacji.

Przez wiele lat ludzie obsługujący komputery i posługujący się nimi nie zwracali większej uwagi na formę czy elegancję rezultatów pracy komputera. Znacznie ważniejszym problemem była szybkość drukowania rezultatów niż ich graficzna jakość. Zresztą dodajmy to na usprawiedliwienie konstruktorów drukarek — żądano od nich początkowo tylko maksymalnej prędkości druku wielometrowych płacht zapełnionych liczbami.

Zasadniczy przełom wywołało pojawienie się komputerów osobistych. Wprawdzie pierwsze małe i tanie drukarki również posługiwały się bardzo uproszczonym, symbolicznym alfabetem, a litery budowane z wyraźnie widocznych rzadkich punkcików daleko odbiegały od jakości normalnych maszyn do pisania, to maniacy komputerowi już wówczas zaczęli pisać na nich swoje pisma i listy. Kiedy komputery osobiste szeroką falą zalały amerykańskie biura i urzędy, zaczął się wyścig także w jakości drukowanych pism. Powstawały coraz doskonalsze drukarki igłowe, strumieniowe i wreszcie laserowe. I te ostatnie właśnie w połączeniu z komputerem osobistym stały się zaczynem procesu, który bez wielkiej przesady nazwać można rewolucją w drukarstwie.

Komputery weszły wprawdzie do poligrafii już przed kilkunastu laty, ale były to nadzwyczaj kosztowne, rozbudowane systemy, na które mogły sobie pozwolić tylko wielkie koncerny prasowe i wydawnicze. Technika druku, od Gutenberga poczynając, opierała się na zasadzie odbijania na papierze przygotowanych uprzednio czcionek. Przez blisko 400 lat w technice druku niewiele się zmieniło, prawie wszystkie prace były ręcznie wykonywane. Dopiero w 1812 r. Friedrich Koenig wpadł na pomysł, aby płaską prasę do odciskania druków zastąpić walcem, co umożliwiło wynalazek maszyny rotacyjnej i znacznie przyspieszyło proces drukowania. Równolegle zaczęto udoskonalać składanie odlewanych ołowianych czcionek. W 1863 roku niemiecki zegarmistrz i wynalazca Otmar Mergenthaler skonstruował pierwszy linotyp, czyli maszynę, która pozwalała odlewać za jednym zamachem cały wiersz. Ten pomysł wystarczył na następne sto lat, i do dziś jeszcze w nielicznych krajach świata (także i u nas) używa się linotypów w drukarniach gazetowych.

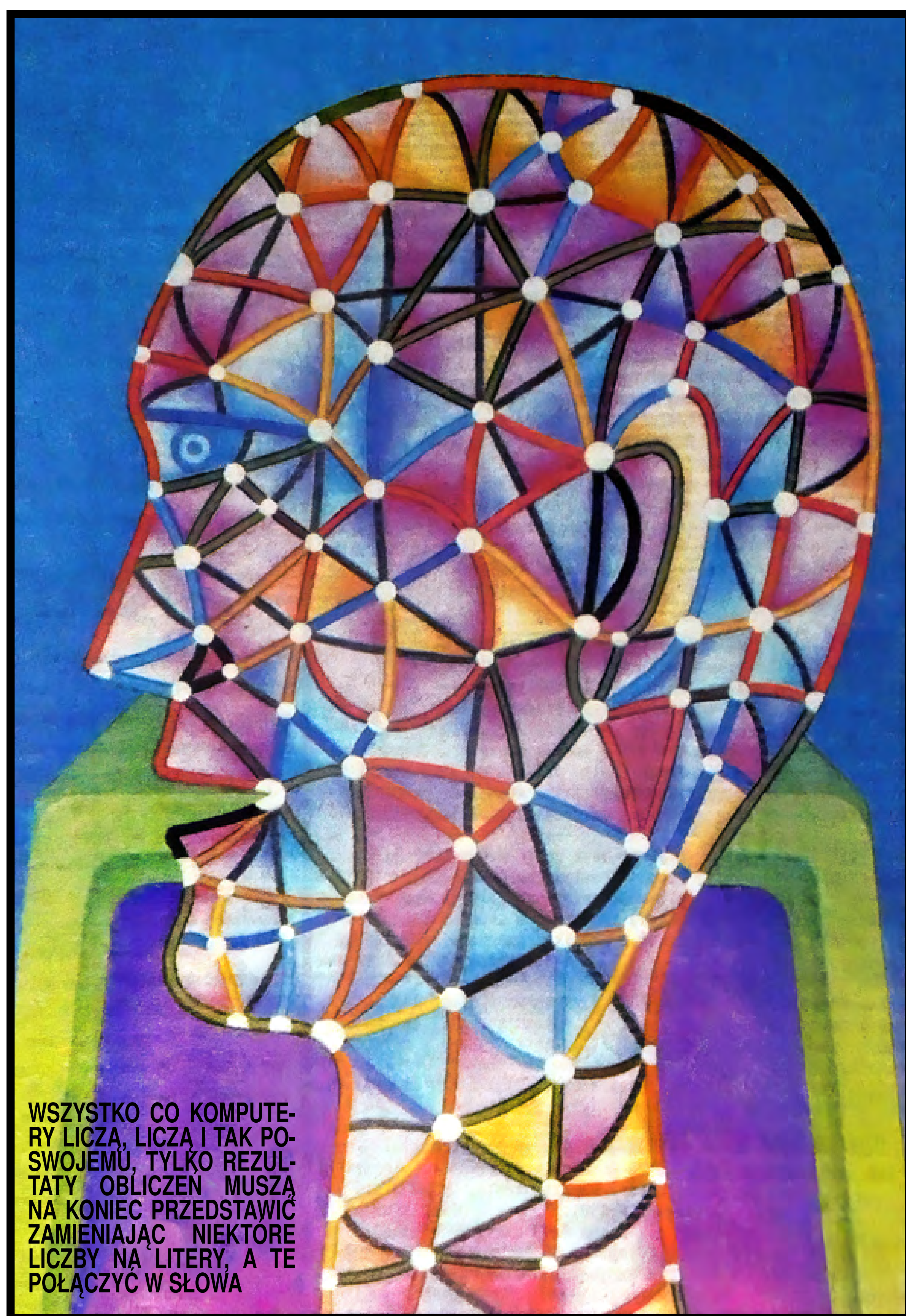
Nowy etap w technice drukarskiej przyniosło zastosowanie techniki tzw. fotoskładu. Początkowo wyświetlano litery tradycyjnymi metodami fotograficznymi, a od czasu gdy pojawiły się tanie lasery, można było już zbudować sterowaną komputerem naświetlarkę laserową, która z wielką prędkością mogła „narysować światłem" całą stronicę książki czy gazety. Ale nadal są to duże, drogie maszyny profesjonalne.

Przed czterema laty, a dokładniej w styczniu 1985 r. po raz pierwszy w fachowej prasie komputerowej

pojawił się termin „Desktop Publishing”. Wówczas to firma Apple wprowadziła do sprzedaży zestaw do prac wydawniczych zawierający nadspodziewanie tanią drukarkę laserową, opartą na konstrukcji japońskiej firmy Canon oraz oprogramowanie specjalne do tworzenia stron „Page Maker” na komputerze Macintosh. Program ten umożliwiał nawet zupełnemu laikowi wykonanie tego, z czym dotychczas musiał biegać do zawodowej drukarni. Za 12 tysięcy dolarów mógł postawić na swoim biurku urządzenie, które zastępowało maszyny do fotoskładu i elektronicznego montażu kosztujące setki tysięcy dolarów. Nic więc dziwnego, że Desktop Publishing stał się zwrotem tak często używanym, że wkrótce operowano już tylko skrótem DTP, a w kilka miesięcy później w USA powstały już pierwsze czasopisma, poświęcone tej gałęzi zastosowań mikrokomputerów.

Co dziś wypada wiedzieć o DTP? Najkrócej mówiąc jest to zestaw urządzeń umożliwiających indywidualnemu użytkownikowi przygotowanie i wydrukowanie publikacji na poziomie technicznym porównywalnym z jakością profesjonalnego składu drukarskiego. Potrzebny jest do tego komputer klasy IBM, jeden z programów wydawniczych (a jest ich już kilkadziesiąt) i drukarka laserowa. Na ekranie monitora (są już specjalne monitory do DTP) autor może dobrać rodzaj i wielkość czcionki, skomponować stronę swojej publikacji, podzielić tekst na szpalty odpowiedniej szerokości, włączyć w tekst wykresy czy tabele skonstruowane także przy użyciu komputera, albo urozmaicić stronę zdjęciami wprowadzonymi do pamięci komputera przy użyciu tzw. scannera. Autor ma przy tym możliwość łatwego wprowadzenia wszelkiego rodzaju korekt i poprawek, a ich skutki są natychmiast widoczne na ekranie.

Po wykreśleniu lub dodaniu słowa komputer sam przebudowuje dalszy ciąg tekstu tak, by zachowane były odstępy między słowami. Gdy efekt na ekranie jest zadawalający, rezultat pracy można wydrukować na własnej drukarce laserowej, która ma rozdzielczość 300 punktów na cal (to znaczy, że drukuje z dokładnością ok. 0,1 mm). Współczesne drukarki laserowe mają wydajność od kilku do kilkunastu stron formatu A-4 na minutę, ale fachowcy przewidują, że w ciągu kilku lat zbudowane będą nowe generacje urządzeń laserowych, które drukować będą dwa razy dokładniej, sto razy szybciej i do tego w kolorze.



WSZYSTKO CO KOMPUTERY LICZĄ, LICZĄ I TAK POSWOJEMU, TYLKO REZULTATY OBLICZEŃ MUSZĄ NA KONIEC PRZEDSTAWIĆ ZAMIENIAJĄC NIEKTÓRE LICZBY NA LITERY, A TE POŁĄCZYĆ W SŁOWA

Dzisiaj, jeśli komuś jakość druku z własnej drukarki nie wystarcza, może zanieść dyskietkę do profesjonalnej drukarni, gdzie przygotowana w domu publikacja zostanie naświetlona na profesjonalnej naświetlarce z dokładnością 2000 punktów na cal i wydrukowana w dużym nakładzie. Można więc przewidywać, że rozwój techniki DTP w przyszłości potoczy się dwiema drogami. Publikacje o niewielkim nakładzie i nie najwyższych wymaganiach technicznych drukować się będzie samemu w domu lub biurze; profesjonalne drukarnie przyjmować będą natomiast nie maszynopisy jak to jest dzisiaj, lecz przygotowane przez autorów i wydawców gotowe do naświetlania strony gazet i książek, przekazywane zresztą przez telefon (u nas pewnie szybciej będzie zawieźć dyskietkę). W obu wypadkach publikacja przygotowana będzie przy użyciu DTP.

Napisano już wiele opowiadań SF malując wizję świata bez papieru. Inspiracją tego kierunku myślenia było sławne określenie amerykańskiego futurologa McLuhana: „elektroniczna, globalna wioska”. Miał to być świat telewizyjnych i komputerowych ekranów, w których nie ma innych środków przekazu wiadomości jak elektroniczne. Tymczasem za sprawą komputerów i DTP przyszłość naszej cywilizacji może wyglądać zgoła inaczej niż sądzili do niedawna futurolodzy. Człowiek jednak lubi wziąć do ręki kawałek zadrukowanego papieru, choćby dlatego, by oderwać się na chwilę od ekranu...

Jan Rurański



DTP NA ATARI

Coraz popularniejszą ostatnio dziedzinę wykorzystania komputerów, jaką jest Desktop Publishing, reprezentuje na Atari zestaw firmy XLEnt Software powstały już w 1985 roku. Jego zalety okazały się tak duże, że wykonano także wersję dla Atari ST.

W skład zestawu wchodzą cztery programy: Page Designer, Rubber Stamp, Typesetter i Megafont II+. Można je oczywiście używać oddzielnie, lecz zastosowane łącznie mają znacznie większe możliwości. Część z nich jest widoczna na wydruku, który został wykonany przy pomocy tych właśnie programów.

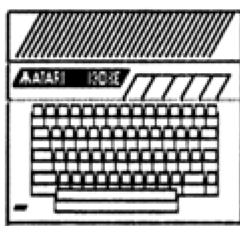
Page Designer (projektant strony) pozwala na rozplanowanie na ekranie monitora wyglądu strony wydruku na arkuszu formatu A4. Na projektowanej stronie można umieścić gotowy rysunek wykonany jednym z pozostałych programów albo przy pomocy Koala Microillustrator. Można także rysować bezpośrednio na projektowanej stronie. Po rozmieszczeniu rysunków wpisujemy tekst. Dostępne są tu dwa tryby pracy: 40- i 80-kolumnowy. W trybie 40-kolumnowym dysponujemy różnorodnymi możliwościami kształtowania tekstu: wybór różnych krojów i wielkości, negatywy, nadruki itd. Przy osiemdziesięciu kolumnach wydruk jest znacznie mniej wyraźny, mniejsze są także możliwości manipulowania tekstem, lecz za to można zmieścić dwukrotnie więcej informacji. Zaprojektowaną stronę można zapisać na dyskietce lub od razu wydrukować.

Rubber Stamp (gumowy stempel) służy do tworzenia skomplikowanych rysunków zawierających tekst i grafikę oraz zajmujących cały ekran lub jego ćwiartkę. Posiada ogromne możliwości przekształcania tworzonej grafiki, dając duże pole do popisu każdemu twórcy. Program pracuje w ósmym trybie graficznym i pozwala na stosowanie dowolnych krojów pisma i dowolne łączenie tekstu z grafiką. Utworzone obrazy można wydrukować albo zapisać na dyskietce w całości lub poszczególnymi ćwiartkami.

Typesetter (zecer) jest programem pozwalającym na wykończenie wykonywanej strony. Posiada on dwie wersje - 65XE i 130XE - różniące się zestawem dostępnych poleceń i wielkością tworzonej strony. Można używać ten program samodzielnie, tworząc na nim od początku tekst i grafikę. Znacznym ułatwieniem w pracy jest jednak możliwość wykorzystania gotowych stron z Page Designer'a oraz rysunków Rubber Stamp'a. Program zawiera też szkicownik do rysunków o rozmiarze 1/4 ekranu (można tu wykorzystać ćwiartki rysunków Rubber Stamp'a). Gotową stronę można wydrukować z rozdzielczością 704 na 624 punkty (Typesetter 65XE) lub 768 na 672 punkty (Typesetter 130XE).

Megafont II+ (drukarz) uzupełnia pozostałe programy umożliwiając dokonywanie dowolnych wydruków na większości drukarek (oprócz bardzo prostych, jak Atari 1029 i Seikosha GP-500). Poza drukowaniem obrazów Rubber Stamp'a i stron Page Designer'a pozwala na skopiowanie dowolnych rysunków wykonanych w grafice 8 lub 15 i zapisanych w formacie nieskondensowanym. Bardzo ciekawą możliwością jest wydruk plików tekstowych zapisanych przez dowolny edytor tekstu (zgodny z formatem DOS 2.0). Można przy tym stosować dowolny krój liter, jeżeli drukarka ma wystarczającej wielkości bufor. Dodatkowym uzupełnieniem jest tryb maszyny do pisania: tekst wprowadzony z klawiatury jest drukowany każdorazowo po naciśnięciu klawisza RETURN.

Opisywany zestaw programów nie jest pierwszej świeżości i ma niewątpliwie różne wady. Jego ogromne możliwości rekompensują jednak niewielkie braki i niedociągnięcia. Należy spodziewać się, że rosnąca popularność Desktop Publishing zaowocuje nowymi, jeszcze lepszymi programami przeznaczonymi także dla ośmiobitowych komputerów Atari.



Wojciech Zientara

Opracowanie: komputer Atari 130XE i zestaw programów firmy XLEnt (Page Designer, Rubber Stamp, Typesetter, Megafont II+)
Wydruk: drukarka Citizen 120-D

NASI TEŻ POTRAFIĄ

Coraz częściej poruszany jest w prasie - nie tylko komputerowej - problem ochrony prawnej oprogramowania. To, co dzieje się w majestacie prawa na różnych giełdach i w "wypożyczalniach" woła o pomstę do nieba.

Dotyczy to nie tylko programów przywiezionych z zagranicy, także twórczość rodzimych informatyków jest kopiowana i sprzedawana bez żadnego poszanowania praw autorskich. Taka sytuacja zniechęca do pisania programów i coraz bliżej mamy do sytuacji, w której - jak napisał w "Komputerze" Andrzej Kadlof - udoskonalane będą tylko programy kopiujące.

Mimo to powstają w Polsce ciekawe programy i to na poziomie profesjonalnym. Chcąc spopularyzować tę dziedzinę twórczości i przyczynić się do usatysfakcjonowania polskich programistów rozpoczynamy cykl, w którym będziemy opisywać napisane w naszym kraju programy dla różnych komputerów domowych.

EUROTEKST

Pierwszy opisywany program zainteresuje przede wszystkim osoby, które wykorzystują komputer głównie jako inteligentną maszynę do pisania. Każdy wie, że komputer stosunkowo łatwo nauczyć pisania polskimi literami. Lecz przekazanie tej umiejętności drukarce jest znacznie trudniejsze. Stosowane są w tym celu rozmaite rozwiązania.

Edytor tekstu "EuroTekst" wykorzystuje do drukowania tryb graficzny. Dzięki temu można nie tylko uzyskać polskie litery na dowolnej drukarce mozaikowej, ale także stosować różnorodne kroje pisma. Oczywiście wpływa to negatywnie na szybkość druku, ale nie można mieć wszystkiego naraz.

"EuroTekst" na pierwszy rzut oka bardzo przypomina popularnego "SpeedScripta". Podobieństwo jest jednak pozorne. Program został wzbogacony o kilka funkcji, m. in. justowanie w prawo oraz podzielony na dwie wersje, które dostarczane są na dwóch stronach dyskietki. Wersja "EuroTekst 4.0" jest edytorem o pełnych możliwościach redakcyjnych. Nie ustępuje ona profesjonalnym programom tego typu, a niektóre przewyższa (np. "SpeedScript").

Wersja "EuroTekst 4.1" została pozbawiona części funkcji redakcyjnych na korzyść wzbogacenia możliwości druku. Umożliwia ona zmianę sterownika (drivera) drukarki, a więc dostosowanie programu do posiadanego sprzętu. Zawiera także funkcję kopiowania rysunków, co pozwala na łączenie tekstu i grafiki. Dodatkowo do tej wersji został dołączony moduł tworzenia własnych zestawów czcionek, dzięki czemu amatorzy wyrafinowanego liternictwa będą mogli zaspokoić wszystkie swoje zachcianki.

Dyskietka z programem zawiera oczywiście sterowniki popularnych drukarek. Są tu uwzględnione m. in. drukarki Star i Atari 1029. Są również gotowe zestawy znaków, które można wykorzystać w pisanych dokumentach. Kilka z nich zostało pokazane na tym wydruku.

Программа "ЕуроТекст" может тоже писат по руски.

Ilustracją możliwości "EuroTekstu" może być ten opis, który w całości (wraz z tytułowymi winietami) został wykonany przy pomocy tego programu oraz komputera Atari 800XL i drukarki Citizen 120-D.

Autor: A.L. System

Dystrybutor: Eurobit, Warszawa

Wojciech Zientara

PROSTA ANIMACJA

Od roku jestem posiadaczem i użytkownikiem komputera ATARI 800 XL i wpadłem na pomysł ciekawej i łatwej animacji. Trybem pracy animacji jest tryb GRAPHICS 10.

Na początku należy zaprojektować 5 faz ruchu postaci, którą chcemy przedstawić. Następnie każdą z faz „rysujmy" na ekranie komputera (każdą fazę innym kolorem) kolorami 4-8. Faza w kolorze 8 będzie na razie niewidoczna, lecz to nie szkodzi. Następnie należy wszystkie fazy „zgasić" instrukcjami POKE 708,0; POKE 709,0: POKE 710,0; POKE 711,0. Po „zgaszeniu" trzeba każdą z nich po kolei „zapalać" i „gasić". Szybkie wykonanie tej czynności uzyskujemy przez bezpośrednie wpisanie odpowiednich wartości do komórek pamięci o numerach 708-712 za pomocą instrukcji POKE. Liczbę, którą należy wpisać otrzymujemy ze wzoru:

KOLOR = KOD KOLORU*16 + POZIOM JASNOŚCI

KODY KOLORÓW:

szary	0
jasnopomarańczowy	1
pomarańczowy	2
czerwono-pomarańczowy	3
różowy	4
purpurowy	5
purpurowo-niebieski	6
niebieski	7
niebieski	8
jasnoniebieski	9
turkusowy	10
zielono-niebieski	11
zielony	12
żółto-niebieski	13
pomarańczowo-zielony	14
jasnopomarańczowy	15

Na przykład gdy mamy takie pięć faz:

kolor fazy	COLOR4	COLOR5	COLOR6	COLOR7	COLOR8
nr komórki	708	709	710	711	712
nr fazy	f1	f2	13	14	15



to instrukcją POKE 708,15 zapalamy f1 kolorem białym, robimy krótką przerwę w wykonywaniu programu instrukcją FOR PRZERWA= 1 TO P:NEXT PRZERWA (liczba P jest długością przerwy), następnie gasimy f1 instrukcją POKE 708,0 i zapalamy f2 instrukcją POKE 709,15 itd. aż do komórki 712.

```
LY   10 GRAPHICS 10
HY   20 COLOR 4:PLOT 40,150:DRAWTO 30,80
ON   30 COLOR 5:PLOT 40,150:DRAWTO 35,80
JZ   40 COLOR 6:PLOT 40,150:DRAWTO 40,80
QO   50 COLOR 7:PLOT 40,150:DRAWTO 45,80
MA   60 COLOR 8:PLOT 40,150:DRAWTO 50,80
UW   69 REM ** GASZENIE FAZ **
EH   70 POKE 708,0:POKE 709,0:POKE
        710,0:POKE 711,0
SG   79 REM ** ZAPALANIE I GASZENIE **
NU   80 POKE 708,15:FOR PRZERWA=1 TO
        40:NEXT PRZERWA:POKE 708,0
QK   90 POKE 709,15:FOR PRZERWA=1 TO
        40:NEXT PRZERWA:POKE 709,0
IP   100 POKE 710,15:FOR PRZERWA=1 TO
         40:NEXT PRZERWA:POKE 710,0
LI   110 POKE 711,15:FOR PRZERWA=1 TO
         40:NEXT PRZERWA:POKE 711,0
OB   120 POKE 712,15:FOR PRZERWA=1 TO
         40:NEXT PRZERWA:POKE 712,0
NY   130 GOTO 80:REM PETLA BEZ KONCA
```

A oto przykładowy program rysujący poruszający się drąg: Oczywiście kształt postaci można dowolnie zmieniać, jednak linie: 70:120 pozostają bez zmian.

Tomasz Banachowicz

ZOSTAŃ NIEŚMIERTELNYM!

Otrzymujemy od Czytelników wiele listów z prośbami o podanie instrukcji POKE umożliwiających uzyskanie „nieśmiertelności" w grach na Atari. Nie jest to tak proste jak w innych typach komputerów, lecz mimo to postaramy się spełnić te życzenia.

Programy dla Atari uruchamiają się automatycznie, więc poprawek nie można wykonywać po wczytaniu programu. Trzeba dokonać zmian w programie przed jego uruchomieniem, czyli na taśmie lub dyskietce. W przypadku programu na dyskietce sprawa jest stosunkowo prosta. Takiej zmiany można dokonać korzystając z któregoś z popularnych monitorów dyskowych: Disk Wizard, Disk Scanner, Sherlock itp.

Znacznie więcej kłopotów sprawiają programy kasetowe. Należy wczytać program przy pomocy jakiegoś monitora, dokonać w nim zmian i ponownie nagrać na kasetę. Można w tym celu wykorzystać np. program „Zamiana napisów" zamieszczony w „Bajtku" 7/87 lub „Nie tylko dla graczy" („Bajtek" 3/88). A teraz czas na poprawki.

Green Beret.

Aktualna liczba „żyć" jest przechowywana w komórce. $06CC. Należy odszukać w programie instrukcję DEC $06CC czyli ciąg bajtów $CE, $CC, $06 (w kodzie ASCII: „N" w negatywie, „L" w negatywie i „CTRL-F") zamienić na LDA $06CC czyli $AD, $CC, $06 (ASCII: „-" w negatywie i dalej bez zmian). Trzeba pamiętać o wpisywaniu wszystkich trzech bajtów jako ciągu do wyszukania, aby nie zamienić w programie wszystkich instrukcji DEC na LDA, a tylko tą, która nas interesuje.

W komórce $06DD znajduje się liczba strzałów z bazooki. Aby mieć ich nieograniczoną liczbę trzeba zamienić DEC $06DD ($CE, $DD, $06; ASCII: „N" w negatywie, „J" w negatywie, „CTRL-F") na LDA $06DD ($AD, $DD, $06; ASCII: „-" w negatywie i dalej bez zmian).

Drop Zone

Liczba super rakiet jest przechowywana pod adresem $05AD. Aby uzyskać dowolną ich ilość, należy zamienić DEC $05AD ($CE, $AD, $05; ASCII: „N" w negatywie, „-" w negatywie, „CTRL-E") na LDA $05AD ($AD, $AD, $05; ASCII: dwa razy „-" w negatywie i „CTRL-E").

Starquake

Aby uzyskać nieśmiertelność trzeba zamienić instrukcję DEC $D2 ($C6, $D2; ASCII: „F" w negatywie) na LDA $D2 ($A5, $D2; ASCII: "%" w negatywie i „R" w negatywie).

Dowolną liczbę strzałów otrzymamy po zamianie w dwóch miejscach pod rząd DEC $D5 ($C6, $D5; ASCII: „F" w negatywie i „U" w negatywie) na LDA $D5 ($A5, $D5; ASCII: "%" w negatywie i „U" w negatywie).

W tej grze można jeszcze w bardzo prosty sposób (bez ingerencji w program) uzyskać 100 „żyć". Gdy bohater zaczyna błyskać należy nacisnąć SHIFT-P (pauza) i po chwili FIRE w joysticku. Po tej operacji licznik wskaże 00, lecz oznacza to 100 „żyć" i po utracie jednego zobaczymy na liczniku 99.

Tomasz Wiśniewski
Wojciech Zientara

KASOWANIE LINII

Posiadam ATARI 800 XL. Jest to dość dobry komputer, lecz niestety jego BASIC pozostawia wiele do życzenia.

Nurtował mnie problem kasowania linii. Jest to łatwe jeżeli chodzi o 2,3 linie ale jeżeli w programie zawierającym 200 linii jest zawarta ładna melodyjka która zajmuje tylko 30 linii, nikomu nie życzyłbym kasowania 170 linii za pomocą edytora. W ostateczności można wczytać np. Turbo Basic XL i wykorzystać instrukcję DEL. Ja jednak pracuję z magnetofonem i ładowanie Turbo Basica zupełnie mi się nie kalkuluje, napisałem więc własny program, który można dołączyć do innego programu i kasować nim żądane linie.

LINIE 1-4 Wprowadzanie danych do kasowania linii
LINIA 5 Wyłączanie ekranu (przyspiesza działanie programu)
LINIA 8 Główna pętla programu
LINIA 9 Wyświetlanie na ekranie numeru
LINIA 10 Kasowanie linii o numerze podanym w linii 9
LINIA 11 Wyłącza odczyt z ekranu*
LINIA 13 Kasowanie ekranu i listowanie programu

Marcin Bochenek

```
GM   1 ? "PODAJ PRZEDZIAL KASOWANIA"
HO   2 ? "PIERWSZA LINIA ";:INPUT PL
WN   3 ? "OSTATNIA LINIA ";:INPUT OL
RJ   4 ? "ODSTEP MIEDZY LINIAMI ";:INPUT OD
QD   5 POKE 559,0:REM PRZYSPIESZA PRACE
TB   6 ? CHR$(125):POSITION 2,10:? "CONT"
LY   7 FOR X=PL TO OL STEP OD
MM   8 POSITION 2,5:? X
NQ   9 POSITION 2,1:POKE 842,13:STOP
RD   10 POKE 842,12:NEXT X:LIST "E:"
```

SORTOWANIE TABLICY LICZBOWEJ

Problem konieczności uporządkowania liczb według wielkości stanął z pewnością nie raz przed wieloma użytkownikami komputerów. Algorytmów rozwiązujących to zadanie powstało wiele, od powolnego sortowania bąbelkowego, aż po skomplikowane, lecz wielokrotnie szybsze kopcowanie (sortowanie stopowe), czy też „sortowanie szybkie".

Czasami wystarczający jest prosty programik sortujący, najczęściej jednak użytkownika interesuje jak najszybsze uporządkowanie zbioru. Ponieważ istotny dla szybkości działania programu jest nie tylko algorytm, ale i język, jaki został użyty przez programistę, najlepszym rozwiązaniem wydaje się być stworzenie odpowiedniej procedury w asemblerze. Wychodząc naprzeciw oczekiwaniom tych, którzy nie mają jeszcze dostatecznego doświadczenia w programowaniu procesora 6502, napisałem procedurę realizującą w kodzie maszynowym algorytm sortowania przez wstawianie — stosunkowo prosty, ale względnie szybki. Sam algorytm prosty i można znaleźć go w wielu publikacjach, ja skorzystałem z książki „Atari Basic".

Na korzyść stosowania asemblera do rozwiązywania problemów, dla których czas jest parametrem rozstrzygającym, niech świadczy następujące porównanie: sortowanie tablicy dwustu elementowej przeprowadzone w Basicu, Turbo Basicu oraz w kodzie maszynowym dało następujące wyniki czasowe:

Atari Basic	322,2 sek.
Turbo Basic XL	134,2 sek.
Asembler	4.5 sek.

Różnica jest chyba dostatecznie wymowna.

Po tym nieco przydługim wstępie czas na praktyczne skorzystanie z programu. Ponieważ zajmuje on więcej niż jedną stronę pamięci, został umieszczony od adresu 39424 ($9A00), czyli tuż przed początkiem pamięci ekranu (w trybie GRAPHICS 0). Oznacza to, że wywołania grafiki o wysokiej rozdzielczości niszczy nieodwołalnie procedurę. Nie jest ona relokowalna ze względu na kilka odwołań wewnętrznych, choć oczywiście po przeróbkach można program umieścić w Innym miejscu. Można też umieścić procedurę w innym miejscu, korzystając z zamieszczonego programu źródłowego.

Wywołanie sortowania wykonuje się przez USR(39424, TYP), gdzie TYP=0 dla porządku wzrastającego, zaś TYP=1 dla malejącego (podawanie innych wartości może spowodować „zawieszenie się" systemu). Sortowana jest pierwsza znaleziona tablica jednowymiarowa, zaś jako parametr powrotny przekazywana jest liczba sortowanych elementów (lub 0, gdy nie znaleziono żadnej tablicy). Aby maksymalnie skrócić działanie programu, na czas obliczeń wyłączane jest DMA dla ANTIC-a, gaśniecie ekranu jest więc normalnym objawem działania programu.

UWAGA: Element tablicy o indeksie zerowym NIE bierze udziału w sortowaniu.

Andrzej Biazik

LISTING 1

```
NF  10 A=39424:FOR I=0 TO 37
RJ  20 S=0:FOR J=0 TO 9:READ B:POKE
       A+J, B:S=S+B:NEXT J
VC  30 READ SK:IF S<>SK THEN 60
RE  40 A=A+10:NEXT I
HU  50 ? " GOTOWE":END
UO  60 ? "  BLAD W LINII ";100+10*I
IY  70 LIST 100+10*I:END
ZG  100 DATA 104,104,104,10,10,10,1
        0,10,105,16,483
KU  110 DATA 141,184,154,165,134,13
        3,203,165,135,133,1547
TV  120 DATA 204,160,0,177,203,201,
        65,240,33,24,1307
GR  130 DATA 165,203,105,8,133,203,
        165,204,105,0,1291
FL  140 DATA 133,204,56,165,204,197
        ,137,144,228,165,1633
OU  150 DATA 203,197,136,144,222,16
        9,0,133,212,133,1549
DG  160 DATA 213,96,160,6,177,203,2
        01,1,208,215,1480
WY  170 DATA 169,0,141,47,2,160,2,2
        4,177,203,925
QI  180 DATA 101,140,133,112,200,17
        7,203,101,141,133,1441
ZB  190 DATA 113,200,177,203,133,20
        7,200,177,203,133,1746
EL  200 DATA 208,24,165,112,105,12,
        133,36,165,113,1073
DP  210 DATA 105,0,133,37,169,2,133
        ,203,169,0,951
CN  220 DATA 133,204,56,165,203,233
        ,1,133,205,165,1498
ZV  230 DATA 204,233,0,133,206,56,1
        65,36,133,40,1206
CM  240 DATA 233,6,133,38,165,37,13
        3,41,233,0,1019
UP  250 DATA 133,39,160,5,177,36,15
        3,114,0,136,953
OU  260 DATA 16,248,162,5,181,114,1
        49,212,202,16,1305
FY  270 DATA 249,160,5,177,38,153,2
        24,0,136,16,1158
BK  280 DATA 248,32,45,155,16,47,16
        0,5,177,38,923
SK  290 DATA 145,40,136,16,249,56,1
        65,38,133,40,1018
RY  300 DATA 233,6,133,38,165,39,13
        3,41,233,0,1021
FY  310 DATA 133,39,198,205,208,202
        ,198,206,16,198,1603
RD  320 DATA 24,165,112,105,6,133,4
        0,165,113,105,968
JG  330 DATA 0,133,41,160,5,185,114
        ,0,145,40,823
WV  340 DATA 136,16,248,24,165,36,1
        05,6,133,36,905
TB  350 DATA 165,37,105,0,133,37,23
        0,203,208,2,1120
EV  360 DATA 230,204,56,165,204,197
        ,208,176,3,76,1519
IT  370 DATA 122,154,56,165,203,197
        ,207,176,3,76,1359
UA  380 DATA 122,154,56,165,207,233
        ,1,133,212,165,1448
PL  390 DATA 208,233,0,133,213,169,
        34,141,47,2,1180
NL  400 DATA 96,56,165,212,197,224,
        240,33,8,104,1335
DW  410 DATA 41,1,72,165,212,16,12,
        165,224,16,924
JN  420 DATA 4,104,106,106,96,104,1
        69,128,96,165,1078
EQ  430 DATA 224,16,2,104,96,104,73
        ,1,76,66,762
SJ  440 DATA 155,162,1,248,56,181,2
        12,213,224,208,1660
GI  450 DATA 13,232,224,6,208,244,1
        65,212,48,10,1362
JV  460 DATA 169,0,216,96,176,246,1
        65,212,48,246,1574
OQ  470 DATA
    169,128,216,96,0,0,0,0, 0,0,609
```

LISTING 2

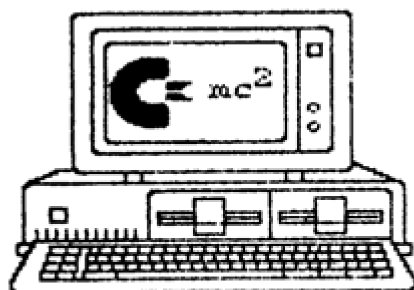
```
0100 *= 39424
0110 FR0 = 212
0120 FR1 = 224
0130 DMA = 559
0140 J = 203
0150 I = 205
0160 N = 207
0170 WTP = 134
0180 STMTAB = 136
0190 STARP = 140
0200 POCZ = 112
0210 BUF = 114
0220 ADJ * 36
0230 ADI = 38
0240 IAD = 40
0250 ;
0260 ; Sortowanie tablicy
0270 ; przez wstawianie
0280 ;
0290 ; (o) 1988 Andrzej Biazik
0300 ;
0310      PLA
0320      PLA
0330      PLA            ; rozpoznanie
0340      ASL A          ; wymaganej
0350      ASL A          ; kolejnosci
0360      ASL A          ; sortowania
0370      ASL A
0380      ASL A
0390      ADC #16
0400      STA ZAP
0410      LDA VVTP       ; przepisz po-
0420      STA J          ; czatek tablicy
0430      LDA VVTP+1     ; wartosci zmien-
0440      STA J+1        ; nych (WTP)
0450      SZU LDY #0
0460      LDA (J),Y      ; szukaj
0470      CMP #65        ; zadeklarowanej
0480      BEQ SPR        ; tablicy
0490      DOD CLC
0500      LDA J          ; gdy nie
0510      ADC #8         ; znalazles -
0520      STA J          ; sprawdz
0530      LDA J+1        ; nastepna
0540      ADC #0         ; zmienna
0550      STA J+1
0560      SEC
0570      LDA J+1
0580      CMP STMTAB+1
0590      BCC SZU        ; sprawdz,
0600      LDA J          ; czy nie zostal
0610      CMP STMTAB     ; przekroczony
0620      BCC SZU        ; obszar VVTP
0630      LDA #0
0640      STA FR0        ; jesli tak - to
0650      STA FR0+1      ; koniec programu
0660      RTS            ; z wartoscia 0
0670      SPR LDY #6
0680      LDA (J),Y      ; sprawdz,
0690      CMP #1         ; czy tablica ma
0700      BNE DOD        ; jeden wymiar
0710      LDA #0
0720      STA DMA        ; wylacz DMA
0730      LDY #2
0740      CLC
0750      LDA (J),Y      ; oblicz
0760      ADC STARP      ; fizyczny adres
0770      STA POCZ       ; poczatku
0760      INY            ; tablicy
0790      LDA (J),Y
0800      ADC STARP+1
0810      STA POCZ+1
0820      INY            ; wpisz do ko-
0830      LDA (J),Y      ; morek N i N+1
0840      STA N          ; liczbe
0850      INY            ; sortowanych
0860      LDA (J),Y      ; elementow+1
0870      STA N+1
0880      CLC
0890      LDA POCZ       ; ustal
0900      ADC #12        ; parametry
0910      STA ADJ        ; poczatkowe
0920      LDA POCZ+1     ; przed
0930      ADC #0         ; wejsciem
0940      STA ADJ+1      ; do petli
0950      LDA #2
0960      STA J
0970      LDA #0
0980      STA J+1
0990      POW SEC        ; poczatek petli
1000      LDA J          ; glownej :
1010      SBC #1         ; J zmienia sie
1020      STA I          ; od 2 do N
1030      LDA J+1
1040      SBC #0
1050      STA I+1
1060      SEC
1070      LDA ADJ        ; korekcja
1080      STA IAD        ; adresow
1090      SBC 6          ; aktualnie
1100      STA ADI        ; sprawdzanych
1110      LDA ADJ+1      ; elementow
1120      STA IAD+1      ; tablicy
1130      SBC #0
1140      STA ADI+1
1150      LDY #5
1160 ET1  LDA (ADJ),Y    ;przechowanie
1170      STA BUF,Y      ;czasowe
1180      DEY
1190      BPL ET1
1200 NAS  LDX #5         ;petla wewnet.
1210 ET2  LDA BUF,X      ;I zmiania sie
1220      STA FR0,X      ;od J-1 do 1
1230      DEX            ;z krokiem -1
1240      BPL ET2
1250      LDY #5         ; przepisz
1260 ET3  LDA (ADI),Y    ; elementy
1270      STA FR1,Y      ; tablicy do
1280      DEY            ; rejestrow
1290      BPL ET3        ; FR0 i FR1.
1300      JSR ODE
1310 ZAP  BPL EXT
1320      LDY #5         ; jesli wynik
1330 ET4  LDA (ADI),Y    ; porownania do-
1340      STA (IAD),Y    ; datni, to wyjdz
1350      DEY            ; z petli
1360      BPL ET4
1370      SEC
1380      LDA ADI        ; korekcja
1390      STA IAD        ; aktualnych
1400      SBC #6         ; wskaznikow
1410      STA ADI
1420      LDA ADI+1
1430      STA IAD+1
1440      SBC #0
1450      STA AD1*1
1460      DEC I          ; zmniejsz
1470      BNE NAS        ; licznik
1480      DEC I+1        ; - jesli sie
1490      BPL NAS        ; nie wyzerowac
1500      CLC            ; to powtorz
1510      LDA POCZ
1520      ADC #6
1530      STA IAD
1540      LDA POCZ+1
1550      ADC #0
1560      STA IAD+1
1570 EXT  LDY #5         ; koniec petli
1580 ET5  LDA BUF,Y      ; wewnetrznej
1590      STA (IAD),Y
1600      DEY
1610      BPL ET5
1620      CLC
1630      LDA ADJ
1640      ADC #6
1650      STA ADJ
1660      LDA ADJ+1
1670      ADC #0
1680      STA ADJ+1
1690      INC J
1700      BNE PRW
1710      INC J+1
1720      PRU SEC        ; sprawdz
1730      LDA J+1        ; czy J<=N
1740      CMP N+1
1750      BCS CZY        ; Jesli tak -
1760      JMP POW        ; to skocz
1770      CZY SEC        ; na poczatek
1780      LDA J          ; petli
1790      CMP N
1800      BCS KON
1810      JMP POW
1820      KON SEC
1830      LDA N          ; Jesli nie -
1840      SBC #1
1850      STA FR0        ; to przepisz
1860      LDA N+1        ; N do FR0
1870      SBC #0
1880      STA FR0+1
1690      LDA #34
1900      STA DMA        ; wlacz DMA
1910      RTS
1920      ODE SEC        ; poczatek
1930      LDA FR0        ; procedury
1940      CMP FR1        ; porownujacej
1950      BEQ ET6
1960      PHP
1970      PLA
1980      AND #1
1990      PHA
2000      LDA FR0        ; sprawdzanie
2010      BPL HP1        ; wielkosci
2020      LDA FR1        ; wyk 1 adn i kow
2030      BPL HP2        ; i znakow
2040      PLA            ; porownywanych
2050 HP4  ROR A          ; i liczb
2060      ROR A
2070      RTS
2080 HP2  PLA
2090      LDA #128
2100      RTS
2110 HP1  LDA FR1        ; wyjscie
2120      BPL HP3        ; z procedury
2130      PLA            ; z ustawionym N
2140      RTS            ; gdy liczba
2150 HP3  PLA            ; Z FR0<FR1
2160      EOR #1
2170      JMP HP4
2180 ET6  LDX #1
2190      SED
2200 ET7; SEC            ; porownywanie
2210      LDA FR0,X      ; kolejnych
2220      CMP FR1,X      ; cyfr
2230      BNE HO2
2240      INX
2250      CPX #6
2260      BNE ET7
2270 HO1  LDA FR0
2280      BMI ET9
2290 ET8  LDA #0
2300      CLD
2310      RTS
2320 HO2  BCS HO1
2330      LDA FR0
2340      BMI ET8
2350 ET9  LDA #128
2360      CLD
2370      RTS
2380      .END
```



NEWSROOM jest programem stosunkowo starym, pochodzacym z 1985 roku i wydanym przez firme Springboard. Byl to jednakze pierwszy program dla C-64 z serii programow zblizonych do Desktop Publishing - umozliwia on zarowno ukladanie tekstu, jak tez mieszanie go z grafika, projektowanie strony (lub stron), ich naglowkow, laczenie fragmentow kolumn w jedna calosc i oczywiscie wydruk w calosci w okreslonym formacie. Program ten oferuje 5 czcionek podstawowych oraz bardzo bogata biblioteke roznych rysunkow i rysuneczkow, ktore mozna wlaczac do tekstu i (co najwazniejsze) modyfikowac za pomoca edytora graficznego o sporych mozliwosciach. Obsluga programu nie jest zbyt prosta (projektowanie strony wymaga ok. 3 godzin dla nowicjusza).

Pojecie DESKTOP PUBLISHING w wypadku Commodore to przede wszystkim laczenie tekstu i grafiki. Takie mozliwosci daje nam szereg nowszych programow: PRINTFOX, PAGEFOX, PRINTFOX PL (opracowany ostatnio w Krakowie przez prywatna firme), a takze GEOS, GEOPUBLISH oraz DESKTOP PUBLISHER firmy TIMEWORKS (dla C-128). Oczywiscie im nowszy jest program tym wiecej mozliwosci oferuje on uzytkownikom.

Pewne funkcje takich programow przejely ostatnio edytory tekstow - m.in. TEXTOMAT 128 czy GEOWRITE umozliwiaja rowniez mieszanie tekstu i grafiki co przy duzej ilosci czcionek umozliwia tworzenie wlasnych "gazetek" z rusunkami wykonanymi z kolei za pomoca edytorow graficznych.



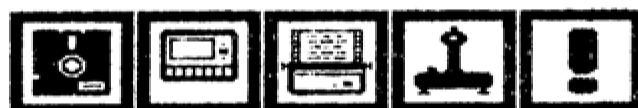
Programy tego rodzaju warto wykorzystywac tam gdzie jakosc nie musi byc pierwszorzedna i tresc ma nad nia przewage. Doskonalym przykladem moga tu byc np. gazetki szkolne, zabawne zaproszenia na bale, biuletyny klubowe, ogloszenia itp.

Oprocz wymienionych powyzej programow istnieje tez cala gama innych, bardziej ukierunkowanych. THE CERTIFICATE MAKER sluzy do wytwarzania dyplomow, laurek czy blankietow ozdobnych; PRINTMASTER PLUS oraz PRINTSHOP sa programami ogolnego przeznaczenia. Niestety niektore z nich (np. NEWSROOM) nie pozwala na stosowanie znakow polskich, chyba ze przy uzyciu roznych trickow.

Pomimo, ze do VENTURY wszystkim tym programom jest daleko, to czy znaczy to, ze sa bezwartosciowe ???

Caly ponizszy tekst i grafika zostaly wykonane za pomoca programu NEWSROOM. Jako jego przeciwienstwo chcialbym takze przedstawic Czytelnikom fragment wydruku wykonanego za pomoca drukarki rozetkowej stanowiacej fragment systemu IBM DISPLAYWRITER SYSTEM. System ten to specjalizowany edytor tekstu umozliwiajacy takze laczenie prostej grafiki (tzn. tabel, wykresow itp.) jednakze o jakosci calkowicie profesjonalnej. No coz, Commodore kosztuje niecale 160 dolarow, natomiast omawiany system - kilkanascie tysiecy..

Choc jest to edytor tekstu o rozszerzonych mozliwosciach, IBM DISPALYWRITER SYSTEM moze byc z powodzeniem wykorzystywany do calkowicie profesjonalnych zastosowan.





PIKTOGRAMY UTWORZONE ZA POMOCA PROGRAMU NEWSROOM

IBM Displaywriter System - rozetka 221

1234567890+q̇wertzuiopńśasdfghjklłąó<yxcvbnm,.-żćęź:;
°*!"#¤%&/()=?QWERTZUIOPŁLKJHGFDSA>YXCVBNM;:_

Czcionka SANS SERIF

Czcionka ENGLISH

Czcionka SERIF SMALL

Czcionka SANS SERIF SMALL

Poniższe twarze zostały wykonane z elementów zapisanych na dyskietkach jako rysunki. Dwa inne doskonale obrazują przysłowie NIE RÓB DRUGIEMU CO TOBIE NIEMIŁE.



PRZEDSTAWIAMY WARSAW BASIC (4)

O PRZERWANIACH I NIE TYLKO

Procesor steruje działaniem mikrokomputera analizując informacje o zachowaniu się systemu. W celu określenia, kiedy zachodzi jakieś zdarzenie (np. wciśnięcie klawiszy STOP(RESTORE), należałoby w sposób ciągły próbkować stan układów we/wy mimo, że względnie mała liczba próbek niesie większość informacji. Podczas jałowego próbkowania profesor jest wykorzystywany nieefektywnie. Efektywność działania można znacznie zwiększyć dopuszczając, aby tylko niosące informację zdarzenia przyciągały uwagę mikroprocesora. Właśnie przerwania umożliwiają budowę takich efektywnych systemów

W C-64 występują następujące typy przerwań: sprzętowe — RESET, NMI i IRQ oraz programowe — BRK. Na konieczność przerwania aktualnie wykonywanego przez mikroprocesor zadania i uruchomienia odpowiedniej procedury obsługującej przerwanie sprzętowe zwraca uwagę procesorowi sprzęt. Przerwanie programowe można wywołać używając odpowiedniej instrukcji języka mikroprocesora 65xx. Sygnałem przerwania sprzętowego dla mikroprocesora jest uziemienie jednej z jego linii przerwań NMI, IRQ lub RESET. Uziemienie powoduje odesłanie do jednej z procedur, których adresy są na stałe zapisane w ROM w komórkach $fffa/$fffb,$fffc/$fffd i $fffe/$ffff.

Przerwanie RESET jest wykonywane m.in. w chwili włączenia C-64 do sieci oraz jest często ostatnią deską ratunku w sytuacji „padnięcia" systemu. Przywraca wtedy ono kontrolę nad systemem bez zerowania pamięci RAM. Jeśli przyczyną „padnięcia" był program w BASICu, to tracimy do niego dostęp z poziomu interpretera, bo wykonuje się automatycznie komenda NEW. Zatem dostęp do takiego programu jest możliwy tylko poprzez bezpośrednie monitorowanie pamięci operacyjnej. Przerwanie IRQ jest wykonywane na gadanie układów we/wy. Procedura obsługująca przerwanie NMT jest wywoływana po wciśnięciu klawiszy STOP/RESTORE. Użytkownik ma możliwość modyfikowania poprzez wektory w Ram procedur przerwań IRQ, NMI i BRK.

Własny system obsługi przerwań ma Warsaw BASIC. Zmodyfikowana procedura IRQ umożliwia podział ekranu monitora na część znakową i część dużej rozdzielczości. Procedura NMI pozwala zachować zmienioną konfigurację systemu, a w szczególności kolory ekranu i kursora, własny zbiór znaków oraz ekran HIRES bez „pluskwy" w postaci kopii części zawartości ROM do RAM. Ponieważ odpowiednie rejestry układów we/wy decydują o częstotliwości wywołań IRQ, to odpowiednie zmiany w tych rejestrach mogą przyspieszyć wykonywanie programu. W Warsaw BASICu tych zmian dokonuje instrukcja SLEEP/SLEEP OFF przyspieszając wykonanie programu o około 10 procent.

W tym odcinku przedstawiamy zmodyfikowaną procedurę NMI.

Zamieszczony obok program 1 zmienia wektor NMI ($03118/$0319) tak, że nowa procedura rozpoczyna się od $c550 i wywołuje podprogram o adresie początkowym $c519, który zachowuje zmodyfikowaną konfigurację systemu. Modyfikacje te obejmują m. in. dobór kolorów ekranu i przeniesienie pamięci ekranu do obszaru o początku na stronie scO. Umożliwia to korzystanie z własnego zbioru znaków, który jest umiejscowiony w RAM pod obszarem ROM zawierającym standardowy generator. Ponadto program 1 zawiera procedurę kopiującą zbiór znaków z ROM do RAM. Dla tych, którzy dołączą program 2 do części publikowanych w poprzednich odcinkach tego cyklu, mamy w następnym odcinku prezent w postaci pokazania, jak w tym systemie tworzyć własne znaki do wyprowadzania na ekran.

Krzysztof Gajewski
Bogusław Radziszewski

```
Program 1
., c4d5 Ida #$c0
., c4d7 sta $0288
., c4da jsr $c519
., c4dd jsr $c508
., c4e0 lda #$d0
., c4e2 ldy #$00
., c4e4 sta $fc
., c4e6 sty $fb
., c4e8 lda ($fb),y
., c4ea sta ($fb),y
., c4ec iny
., c4ed bne $c4e8
., c4ef inc $fc
., c4f1 lda $fc
., c4f3 cmp #$e0
., c4f7 bne $c4e8

., c4f7 pha
., c4f8 lda $01
., c4fa ora #$04
., c4fc sta $01
., c4fe Ida $dc0e
., c501 ora #$01
., c503 sta $dc0e
., C406 pla
., c507 rts

., c508 pha
., c509 lda $dc0e
., c50c and #$fe
., c50e sta $dc0e
., c511 lda $01
., c513 and #$fb
., c515 sta $01
., c517 pla
., c518 rts

., c519 lda #$50
., c51b sta $0318
., c51e lda #$c5
., c520 sta $0319
., c523 lda #$00
., c525 sta $d020
., c528 sta $d021
., c52b Ida #$07
., c52d sta $0286
., c530 lda $dd00
., c533 and #$fc
., c535 sta $dd00
., c538 lda $d011
., c53b and #$fd
., c53d sta #d011
., c540 lda #$00
., c542 sta $d01a
.. c545 lda #$08
., c547 sta $d016
., c54a lda #$06
., c54c sta $d018
., c54f rts

., c550 pha
., c551 txa
., c552 pha
., c553 tya
., c554 pha
., c555 lda #$7f
., c557 sta $dd0d
., c55a ldy $dd0d
., c55d bmi $c570
., c55f jsr $f8bc
., c562 jsr $ffe1
., c565 bne $c570
., c567 jsr $e51b
., c56a jsr $c519
., c56d jmp $e39d
., c570 jmp $fe72
```

```
Program 2
300 print" Czesc 3 "
305 x=50389:n=157:c=0
310 for i=0ton : reada : pokex+i,a:c=c+a:next
315 ifc=19451thenprint" Czesc 3 ok ":goto325
320 printc: print" Blad w danyc czesci 3 ":end
325 sys50389:end
330 data169,192,141,136,2,32,25,197,32,8,197,169
335 data208,160,0,133,252,132,251,177,251,145,251,200
340 data208,249,230,252,165,252,201,224,208,241,72,165
345 data1,9,4,133,1,173,14,220,9,1,141,14
350 data220,104,96,72,173,14,220,41,254,141,14,220
355 data165,1,41,251,133,1,104,96,169,80,141,24
360 data3,169,197,141,25,3,169,0,141,32,208,141
365 data33,208,169,7,141,134,2,173,0,221,41,252
370 data141,0,221,173,17,208,41,223,141,17,208,169
375 data0,141,26,208,169,8,141,22,208,169,6,141
380 data24,208,96,72,138,72,152,72,169,127,141,13
385 data221,172,13,221,48,17,32,188,246,32,225,255
390 data208,9,32,27,229,32,25,197,76,157,227,76
395 data114,254
```

DRUKARNIA W 8 BITACH

Motywem przewodnim tego numeru BAJTKA jest praca nad tekstem przy pomocy komputera. Bardzo wiele programów DTP zajmuje się kompozycją tekstu i grafiki w dokument czy gazetę. W rezultacie otrzymujemy materiał estetyczniej- szy niż wówczas gdy każda z części jest opracowywana oddzielnie np. za pomocą edytora tekstu.

Twórcy takich programów z pogardą patrzą na komputery 8-bitowe twierdząc, że programy na takie zabawki nigdy nie osiągną możliwości pakietu typu Ventura Publisher. Jak się jednak ostatnio okazało, komputery firmy Commodore (C-64 i C-128) doczekały się programu, który daje aż 85 proc. możliwości programu Ventura Publisher (80 proc. ÷ 90 proc. dla innych programów profesjonalnych), a kosztuje o 90 proc. mniej niż takie programy. Programem tym jest GeoPublish firmy Berkeley Softworks, współpracujący z programem (systemem operacyjnym) GEOS. Minimalne wymagania GeoPublish to komputer C-64 lub C-128: program GEOS 1.2 lub późniejszy dla komputerów C-64 lub C-128 (również dla C-128 w trybie 128); stacje dysków 1541, 1571, 1581 (ta ostatnia zostanie przedstawiona niebawem w BAJTKU); joystick lub myszka i drukarka. Proponowanymi przez firmę rozszerzeniami są programy: Desk Pack 1 i GeoWrite Workshop, umożliwiające zwiększenie liczby tytułów programów, z których można przenosić grafikę i tekst, oraz moduł rozszerzający pamięć komputera, dający możliwość stworzenia RAM-dysku przyspieszającego pracę programu (dla C-128) i druga stacja dysków. W tym miejscu muszę dodać, że bez programu Desk Pack 1 nie mamy możliwości przenoszenia zbiorów z konkurencyjnego choć dużo starszego Newsroom'u. Pomimo tego handicapu pozostaje jeszcze sporo tytułów programów współpracujących z GeoPublish. Są to oczywiście elementy ułatwiające wykorzystanie potężnych możliwości programu, ale i bez nich program GeoPublish również działa dając użytkownikowi dostęp do wszystkich swoich możliwości. Cały program i dodatkowe zbiory (np. dodatkowe czcionki do druku o rozdzielczości, bagatela, 192 punkty na 2.5 cala) mieszczą się na dwóch stronach jednej dyskietki i w odróżnieniu od programów profesjonalnych nie wymagają stosowania dysku twardego. Główny program ma objętość 95 KB i składa się z oddzielnych modułów, dzięki czemu nie ma potrzeby wczytywania go w całości do pamięci komputera (zresztą nie ma takiej możliwości w komputerze dysponującym 64 KB pamięci). Mo- dułowość programu przyspiesza jego działanie pozostawiając użytkownikom więcej wolnej pamięci. Częste korzystanie z różnych opcji programu radykalnie zwalnia jego pracę, szczególnie przy współpracy ze stacją 1541, ponieważ program musi doczytywać kolejne części dbając o dane zawarte już w pamięci. Stacje 1571, a szczególnie znacznie szybsza 1581, nie dają powodów do narzekania na wolną transmisję danych.

Jak przystało na każdy porządny DTP, tak i na ten trzeba poświęcić sporo czasu, by opanować podstawy jego obsługi. Podczas jednak gdy na opanowanie funkcji programu Ventura Publisher potrzeba około 3 dni, tu wystarcza 1 dzień. Stało się tak dzięki bardzo dobrze napisanej, obszernej instrukcji

obsługi, która porusza wszystkie aspekty pracy programu (jedyną zauważoną do tej pory wadą tego podręcznika jest brak opisu sposobu instalacji programu w GEOS 128). Lektura jednego tylko rozdziału umożliwia stworzenie własnej gazety w czasie około jednej godziny. Użytkowników, którzy nauczyli się wcześniej korzystać z programu GEOS, czeka trochę mniej pracy w zapoznawaniu się z programem, gdyż mimo wielu nowości GeoPublish zachował cechy pracy kolejnych wersji GEOS-a.

Przejdę teraz do krótkiego scharakteryzowania czterech trybów pracy GeoPublish nazwanych przez autorów: Master Pages, Page Layout, Page Graphics i Editor. I tak: Master Pages umożliwia definiowanie wymiarów strony gazety, czy innej publikacji, określenia ilości kolumn (maksymalnie 4) oraz odstępów między nimi jak również grafiki i tekstów, które będą się pojawiać na każdej lub prawie każdej stronie. Po takim ustaleniu, w dalszej pracy nie jesteśmy skrępowani wstępnymi ustaleniami. W razie potrzeby rozłożenie danej strony może być inne od pozostałych.

Drugim modułem jest Page Layout, czyli, jak kto woli, tryb rozplanowania zawartości stron. Po wstępnych ustaleniach w poprzednim trybie (stworzeniu jak gdyby szkieletu konstrukcji) możemy przystąpić teraz do lego wypełniania tekstami i grafiką — w kolejnych modułach. Jak wspomniałem wcześniej, gotowe materiały mogą pochodzić z innych programów (edytory tekstu i programy graficzne). Nie jest to jednak konieczne, gdyż poszczególne elementy możemy również tworzyć w GeoPublish, choć, prawdę mówiąc, szybsze będzie napisanie artykułu, gdy skorzystamy z innego edytora (są to moje osobiste odczucia). Nakładając dotychczasowe materiały, możemy je w trzecim module wzbogacić o nietypowe elementy graficzne i specyficzne napisy. Dysponujemy możliwością powiększenia i pomniejszania grafiki (choć trwa to nieraz długo), tworzenia ramek, wzorków, szlacz ków... Program zawiera również rewelacyjną w działaniu procedurę rysowania okręgów, elips czy łuków. Niektórzy twierdzą, że takie procedury warte są połowy ceny programu. Po pracy w już wymienionych trybach możemy teraz przejść do ostatniej części programu czyli Edytora. Jest to edytor w dosłownym tego słowa znaczeniu, w odróżnieniu od procesora tekstu! Tryb ten najbardziej przypomina ekran z poprzednich wersji GEOSA (jest idealną kopią ekranu wersji 1.2). Edytor posiada wiele różnych możliwości i funkcji, choć szybko rzuca się w oczy brak takich funkcji jak ZNAJDŹ (SEARCH), czy ZNAJDŹ I ZASTĄP (SEARCH AND REPLACE). Edytor ten jest, jak wspomniałem, bardzo zbliżony do Geowrite, tak więc użytkownicy, którzy pracowali wcześniej z tym programem, nie powinni mieć żadnych trudności.

Jak widać z artykułu, sam program i jego możliwości zostały opisane raczej z myślą o przedstawieniu Czytelnikom informacji, że taki program w ogóle istnieje; lista jego możliwości i wzajemnych zależności zajmuje oczywiście znacznie więcej miejsca aniżeli mamy w klanie Commodore i z tego powodu nie podajemy jej tutaj.

To, co do niedawna było tylko marzeniami, teraz stoi w zasięgu ręki za jedyne 70 dolarów amerykańskich. GeoPublish rzeczywiście udostępnia możliwości profesjonalnego sprzętu i oprogramowania, konkurując z nimi niską ceną, i stanie się on niebawem kolejnym standardem wśród programów tego typu.

Dominik Falkowski

PROGRAMY GRAFICZNE

Spośród programów graficznych na Spectrum godne uwagi są jedynie THE ARTIST i ART STUDIO.

Pierwszy z nich był rewolucją w tworzeniu grafiki. Ukazał się w 1985 roku i jego powodzenie nie słabnie do dziś. Podobnie ART STUDIO — zadziwił użytkowników nawet Macintosha. Spróbujmy przyjrzeć się obu tym programom pod kątem przydatności nie tylko grafikowi (jemu wystarczy pędzel), lecz i zwykłemu użytkownikowi, chcącemu nabazgrać jakiś miły rysunek.

Nie trzeba chyba wyjaśniać, że bariera rozdzielczości 256x192 i 8 kolorów nie została przeskoczona. Pozytywnym natomiast jest fakt, że rysować można na całym ekranie, łącznie z dwoma dolnymi liniami. Należy wówczas przesunąć ekran w dół (ART STUDIO) lub w górę (ARTIST). ART STUDIO robi to jednak dużo bardziej elegancko — wskazujemy kursorem strzałkę w górę (!), (program działa tak, jakbyśmy patrzyli na ekran przez nieco mniejsze okno, trzeba więc je przesunąć w górę). ARTIST załatwia sprawę po wciśnięciu jednego klawisza.

Jak poruszać kursorem po ekranie? Bardzo prosto! W ARTIST służą do tego cztery klawisze i piąty do postawienia punktu lub joystick Kempston. Ciekawostką jest fakt, że program „czuje", czy jest podłączony interface i odpowiednio się przestawia. Natomiast w ART STUDIO mamy możliwość wybrania sterowania — definiowanymi klawiszami, joystickiem lub ... myszą systemu AMX.

Zacznijmy więc od postawienia punktu. ARTIST zrobi to jednym klawiszem. Niedogodnością jest używanie go także do ścierania punktu. Tryb ścieranie — malowanie wybierany jest także wciśnięciem innego klawisza. Kursor ma postać kropki, więc punkt zauważymy dopiero po odsunięciu go na bok.

Kursor ART STUDIO to strzałka. Aby postawić punkt na ekranie musimy wybrać opcję POINTS z okna SHAPES. Kursor zmieni się na „celownik" i teraz — hulaj dusza!

Bardzo dobry jest pomysł zmiany kursora, tzn. na ekranie jest on taki, jaki jest, natomiast w górnej części ekranu (menu) ma zawsze postać strzałki.

Nie ma co dalej rozwodzić się nad rysowaniem figur — linie, trójkąty, czworokąty, okręgi, łamane są uzyskiwane w podobny sposób. Dużą zaletą ART STUDIO jest obecność opcji ELASTIC. Pozwala ona na rozciąganie rysowanej figury zanim się ją jednoznacznie wybierze (np. po zaznaczeniu początku linii, jest on łączony z kursorem i dopiero po wciśnięciu FIRE połączenie to staje się linią).

Następna ważna sprawa — przybliżanie rysunku. ARTIST to znów jeden klawisz powiększający otoczenie kursora ok. cztery razy. W ART STUDIO musimy wybrać okno MAGNIFY a z niego skalę — 2,4 lub 8 razy. Kursor zmienia się w lupę i po wciśnięciu FIRE powiększa dany obszar. Powiększone okno możemy przesuwać, pokryć siatką lub zmieniać. Tu postawienie punktu (nie ma wyboru) odbywa się przez wciśnięcie FIRE w danej kratce. Można go również zmazać — do wybrania trybu służą opcje SET, RESET i TOGGLE.

Teraz kolory. ARTIST znów nie daje nic do myślenia — PAPER, INK ustawiane są odpowiednimi klawiszami. Natomiast praca z ART STUDIO stwarza ogromne wrażenie. Wybieramy opcję np. INK z okna ATTRIBUTES i wskazujemy kursorem kwadracik w odpowiednim kolorze. Podobnie ustawiamy BRIGHT, INVERSE i OVER. Nie możemy oczywiście mieszać kolorów w kwadracie znakowym (impresjonista umarł by z żalu), ale podobny program na Macintosha oferuje tylko dwa kolory — czarny i biały!

Ważne jest opatrywanie rysunków komentarzem słownym. W ARTIST na ekranie pojawia się kwadratowy kursor, przy pomocy którego możemy wpisać dowolny tekst. Używamy czcionki normalne] wielkości lub zmniejszonej. Można także mieszać kroje pisma, zdefiniowane z góry w pamięci. ART STUDIO daje szerokie możliwości — znaki do potrójnej wysokości (szerokości), pisane od lewej do prawej i z góry na dół. Można także bardzo prosto zdefiniować nowy zestaw znaków (FONT).

A inne techniki malarskie? Istnieją tylko w ART STUDIO. Możemy wybrać np. pędzel o definiowalnym wzorze i postawić nim kilka ryjków na ekranie. Możemy wziąć SPRAY o wybranej gęstości i zapylić dany fragment ekranu. Oba programy dysponują oczywiście zapełnianiem obszaru wybranym (lub zdefiniowanym) wzorem.

Podsumowując, oba opisane programy prezentują wysoką klasę programów użytkowych, o ogromnych możliwościach. Nie opisałem tu okien, drukowania ekranu na drukarce, animacji (tak! to ARTIST) wybranych fragmentów obrazu, i innych przydatnych opcji. Oba programy warte są uwagi, na pewno nie do ilustracji gazety, lecz do własnych potrzeb są doskonałe.

(mp)



EDYTORY TEKSTU

Niemało jest edytorów tekstu napisanych specjalnie na Spectrum.

Spośród nich warto wymienić: Tasword Two, Tasword Three, PolTasword, The Last Word, The Writer, Tekst Edytor oraz CX Tekst. Wszystkie oferują usługi, jakich wymaga się od profesjonalnych edytorów na profesjonalnych komputerach, jak np. WordStar na sprzęt typu PC. Mamy więc marginesy, odnośniki, przenoszenie słów, znaki specjalne i wiele innych. Jedynym mankamentem wszystkich tych edytorów jest ograniczenie tekstu wpisanego naraz do około 300 linii.

Skoncentrujmy się więc nad dwoma spośród nich. Najlepiej wybrać THE WRITER (inny produkt tej firmy to THE ARTIST) — nowatorski pomysł w tej trudnej dziedzinie, jaką jest przetwarzanie tekstów na Spectrum oraz POLTASWORD — bardzo zręczną przeróbkę edytora TASWORD TWO dla potrzeb polskiego literata, dokonaną przez Tadeusza Wilczka.

Podstawową rzeczą, dla której wolę POLTASWORD. jest obecność polskich znaków, dostępnych w bardzo prosty sposób. Można też, co prawda, zdefiniować je i w THE WRITER, ale nie będą tak łatwo dostępne.

THE WRITER odparowuje jednak cios łatwością modyfikacji samego siebie. Wystarczy wcisnąć BREAK w odpowiednim momencie i już jesteśmy w normalnym Basic-u Spectrum. Można teraz zmienić, co się tylko zamarzy, na przykład umożliwić edytorowi współpracę z microdrive lub stacją dysków. Potem tylko RUN lub CONTINUE i można tworzyć dalej.

Inną zaletą programu THE WRITER, której nie powstydziłby się nawet ChiWriter na PC, jest możliwość rozszerzenia jednej linii tekstu do 100 znaków mimo, że ekran może pokazać jedynie 64 znaki. Część tekstu będzie po prostu niewidoczna. Natomiast na drukarce uzyskamy wydruk takiej szerokości, jakiej szerokości jest tekst. POLTASWORD może, niestety, operować tekstem o szerokości do 64 znaków, lecz mamy tu opcję, pozwalającą uzyskać na wydruku znaki podwójnej wysokości.

Bardzo istotną rzeczą jest komunikacja z użytkownikiem. Najwyżej stoi tu WRITER — wydawanie poleceń odbywa się przez wybranie kursorem jednego ze spisów pojawiających się na środku ekranu. Każdy spis zawiera do kilku opcji służących do operowania tekstem. Ten sposób komunikacji jest dużo prostszy do opanowania niż kilkadziesiąt klawiszy kontrolnych w programie POLTASWORD wciskanych w trybie EXTENDED MODE. Na szczęście możemy tu przywołać na ekran listę wszystkich klawiszy z opisem w języku angielskim. WRITER ma dużo szerszą opcję HELP; informuje ona nie tylko o dostępnych klawiszach, ale i o sposobie posługiwania się edytorem, zawartości spisów, kodach drukarki. Na nieszczęście cała opcja nie mieści się w pamięci i jest doładowywana z magnetofonu lub innego urządzenia zewnętrznego. Powrót do tekstu przed przejrzeniem wszystkich stron możliwy jest przez BREAK i RUN.

Obowiązkowo wszystkie edytory mają opcję SEARCH/REPLACE, pozwalającą na przeszukiwanie tekstu w celu znalezienia ciągu znaków i ewentualne zastąpienie go innym Tekst może być po tym automatycznie formatowany, choć można to wywołać bezpośrednio. Nie brak również operacji na blokach — przesuwania, kasowania, zaznaczania a nawet nagrania na nośnik w celu późniejszego dołączenia do innych tekstów.

Napisany tekst możemy wydrukować na prawie każdej drukarce graficznej. Kłopot jest jedynie z THE WRITER — do zmiany drukarki potrzebny jest program konfigurujący, w którym deklarujemy niezliczoną ilość opcji drukarki.

Niezwykle ciekawą i użyteczną sprawą jest możliwość komunikacji z innymi komputerami przez złącze RS 232. Służy do tego program dołączony do edytora THE WRITER. Umożliwia on transmisję tekstów napisanych nawet na edytorze tekstów WordStar komputera IBM PC i komputerów pracujących w systemie operacyjnym CP/M. Teksty można przesyłać w jednym z trzech formatów — ASCII, WORDSTAR oraz własnym formacie THE WRITER. Możliwe jest więc proste stworzenie sieci komputerowej.

Jak więc widać, urządzenie o nazwie ZX Spectrum nadaje się do przetwarzania tekstów całkiem nieźle i może być z powodzeniem używane do poważnych prac. Znam osobę, która pracę doktorską z dziedziny neurologii pisała właśnie na Spectrum, przy użyciu edytora POLTASWORD.

Marcin Przasnyski

OD ŚRODKA cz. V

Cześć! W dzisiejszym odcinku powiemy sobie jeszcze trochę o BASIC-u. lecz zajmiemy się już także asemblerem, czyli tym, co dzieje się po wykonaniu w loaderze instrukcji RANDOMIZE USR...

Wszystkie gry mają bardzo dobrze zabezpieczoną swą część BASIC-ową, w końcu to najważniejszy (pod względem skuteczności zabezpieczeń) element programu — od BASIC-a przecież zaczyna się wczytywanie całego programu. Jeśli BASIC-owy loader jest słabo zabezpieczony, to włamanie się do całego programu jest znacznie ułatwione, na co przykładem są loadery firmy ULTIMATE, prezentowane miesiąc temu. Jedną z metod łamania zabezpieczeń loaderów jest wczytywanie ich za pomocą programu „load/merge" (patrz drugi odcinek naszego cyklu). Jednak czasem lepiej jest umieścić ten loader nie w pamięci przeznaczonej na BASIC, lecz powyżej RAMTOP-u, by móc oglądać go bez żadnej obawy dokonania w nim przypadkowych zmian.

Istnieje na to bardzo skuteczna metoda — wczytanie programu w BASIC-u, jako blok kodu maszynowego, pod wygodny dla nas adres. Aby tego dokonać, potrzebna jest znajomość długości programu, który chcemy wczytać (możesz użyć procedury „czytacz" z numeru 2/88 Bajtka), choć bez długości także można się obejść. Ponadto potrzebne jest trochę wolnego miejsca na taśmie magnetofonowej. Sposób ten polega na oszukaniu instrukcji LOAD przez podmienienie nagłówków.

Na wolnej taśmie nagrywamy nagłówek bloku kodu, przez SAVE „bas" CODE 30000,750, jeśli wiemy, że długość programu wynosi 750 bajtów. Jeśli nie znamy jej — podajemy odpowiednio większą wartość — nawet rzędu kilkunastu lub kilkudziesięciu kB, choć program może mieć raptem 100 bajtów. Na taśmę nagrywamy tylko sam nagłówek, przerywając później nagrywanie wciśnięciem BREAK. Teraz ustawiamy taśmę tuż przed nagranym właśnie nagłówkiem, a taśmę z programem — tuż za nagłówkiem programu, ale przed właściwym blokiem danych.

Wpisujemy

CLEAR 29999: LOAD „" CODE
lub CLEAR 29999: LOAD „" CODE 30000

i wczytujemy nagłówek. Zaraz po jego wczytaniu wciskamy STOP w magnetofonie, wymieniamy kasety i ponownie wciskamy START (przez cały ten czas komputer czekał na blok danych). Teraz wczytuje się program BASIC-owy, ale pod adres 30000 — powyżej RAMTOP-u. Jeśli podaliśmy w nagłówku zawyżoną długość programu, to wczytywanie skończy się komunikatem „Tape loading error", ale to nie szkodzi — nie oznacza to (najprawdopodobniej) błędu wczytania, lecz właśnie fakt zbyt małej ilości danych niż podana w nagłówku. Teraz już w dowolny sposób możesz oglądać wczytany program, nawet pisząc do tego celu własny program w BASIC-u (nie musi to być jedna linia, do bezpośredniego wykonania, jak dotychczas).

Oprócz tej metody istnieje jeszcze druga, lecz by z niej korzystać, koniecznie trzeba znać asembler (a warto korzystać, bo daje ona większe możliwości łamania programów, a jej znajomość pozwala zwykle rozszyfrować, jak działa program ładujący).

Bardzo często (szczególnie w najnowszych programach) spotykane są bloki programów zapisane i wczytywane do pamięci komputera, bez nagłówka. Jest to dosyć oryginalne i efektowne zabezpieczenie, odstraszające zwykle początkujących, ale złamanie takiego programu wcale nie jest trudne. Cała tajemnica polega na wykorzystaniu znajdujących się w ROM-ie Spectrum procedur, używanych przez instrukcje LOAD, SAVE, VERIFY i MERGE.

Pod adresem 1366 (hex 0556) znajduje się procedura LOAD-BYTES wczytująca z magnetofonu blok danych, czyli pilota i następujące po nim informacje. Nie jest przy tym ważne, czy będzie to nagłówek, czy też właściwy blok danych, które należy umieścić gdzieś w pamięci.

Zacznijmy jednak od początku. Każdy zabezpieczony program rozpoczyna się od dłuższego lub krótszego loadera napisanego w BASIC-u. Program korzystający z ładowania bez nagłówków (przez procedurę 1366 lub inną) musi być napisany w kodzie maszynowym, jak każda procedura obsługująca magnetofon. Najczęściej program ten umieszczany jest w jednej z linii BASIC-a, np. po instrukcji REM, lub w obszarze zmiennych BASIC-a. Po wczytaniu, BASIC-owy ioader uruchamia się i wykonuje instrukcje RANDOMIZE USR ... inicjując tym samym działanie programu maszynowego.

Procedura LOAD-BYTES wymaga odpowiednich parametrów wejściowych. Przekazywane są one w odpowiednich rejestrach mikroprocesora, tak więc w rejestrze IX podajemy adres, pod który chcemy wczytać blok danych, a w parze DE, długość tego bloku. W akumulatorze umieszczamy 0 — jeśli chcemy wczytać nagłówek albo 255 — jeśli ma to być blok danych. Ponadto znacznik przeniesienia (CARRY) ustawiamy na 1, gdyż inaczej zamiast LOAD, procedura 1366 spełniałaby funkcję VERIFY. Oto przykład procedury ładującej z taśmy obrazek bez nagłówka:

```
LD IX, 16384    ;adres wczytania
LD DE,6912      ;dlugosc bloku
LD A,255        ;blok danych SCF ;ustaw CARRY,
                 czy-
CALL  1366      ;li LOAD
RET ;powrot
```

Procedura 1366 w razie błędu wczytania, nie drukuje komunikatu „Tape loading error". Istnieje natomiast druga procedura ładująca, która to robi. Znajduje się ona pod adresem 2050, a wygląda tak:

```
2050 CALL 1366 ; wczytanie bloku danych
2053 RET C ; powrot,  jesli nie bylo
2054 RST 8 ; bledu,   inaczej RST 8 z
2055 DEFB 26       j komunikatem  Tape ...
```

Po powrocie z procedury 1366, wskaźnik przeniesienia zawiera informację o prawidłowości wczytania bloku. Jeśli jest on skasowany, oznacza to, że wystąpił błąd. Niektóre loadery wykorzystują właśnie procedurę 2050, a nie 1366.

Czasem procedury ładujące nie korzystają ani z jednej, ani z drugiej procedury, ale zastępują je własną, jest ona jednak zwykle bardzo podobna do procedury 1366 lub jest wręcz jej przeróbką, dzięki której np. bloki danych ładowane są w dół pamięci — od adresów wyższych do niższych, lub np. z inną prędkością. Taki program należy analizować za pomocą disassemblera (np. MONS-a) porównując niektóre jego fragmenty z tym, co znajduje się w ROM-ie (procedura 1366 była opisana szczegółowo w numerze 8/86 „Komputera").

A oto jak wykorzystać procedury z ROM-u do wczytania BASIC-a pod dowolny adres a nie w obszar dla niego przeznaczony: Najpierw za pomocą „czytacza" odczytujemy nagłówek programu, do którego chcemy się włamać i zapamiętujemy jego długość (tzn. długość całego bloku — programu wraz ze zmiennymi). Teraz wpisujemy odpowiedni program w asemblerze, który wczyta BASIC pod taki adres, jaki chcemy (powyżej RAMTOP-u):

```
LD IX,adres_wczytania
LD DE,dlugosc_bloku
LD A,255 SCF
JP 2050
```

Podobnie jak w sposobie z podmienianiem nagłówków, jeśli nie znamy długości programu — możemy podać wartość zawyżoną, lecz wówczas wczytanie zakończy się komunikatem „Tape loading error". Każdorazowe wczytywanie asemblera po to, by wpisać powyższy program może być denerwujące, lepiej więc tworzyć go z poziomi BASIC-a poprzez POKE-i:

```
10 INPUT  Adres wczytania BASIC-a ?  sa
20 RANDOMIZE a:  CLEAR a-1
30 LET a=PEEK 23670: LET b=PEEK 2 3671:
   LET adr=256*b+a
40 INPUT  dlugosc BASIC-a ? ;c
50 RANDOMIZE c: LET c=PEEK 23670: LET d=PEEK
   23671
60 FOR n=adr TO adr+11
70 READ x: POKE n,x: NEXT n: DATA
   221,33,a,b,17,c,d,62,255, 195,2,8
80 RANDOMIZE USR adr
```

Ustawiamy taśmę z rozpracowywanym programem za jego nagłówkiem. Teraz uruchamiamy powyższy program, podajemy dane i włączamy magnetofon. Skutek jest identyczny jak przy podmienianiu nagłówków, ale pierwszą widoczną zaletą tej metody jest to, że przy okazji nie robimy bałaganu na kasetach.

Na zakończenie wypada wspomnieć o jeszcze jednej procedurze, umieszczonej w ROM-ie pod adresem 1218. Jest to procedura SAVE-BYTES, odwrotna do LOAD-BYTES, tzn. nagrywająca na taśmę blok o podanych parametrach: przed jej wywołaniem w rejestrze IX umieszczamy adres od którego rozpocznie się nagrywanie, DE zawiera długość bloku do wysłania. W akumulatorze zaznaczamy czy ma to być nagłówek (0), czy blok programu (255). Stan wskaźnika CARRY nie jest ważny.

Za miesiąc pokażemy jak omawiane dziś procedury z ROM-u stosowane są w konkretnych programach.

Tomasz Surmacz
Robert Dudzik

Z KLAWIATURY NA JOYSTICK

(cz. 2)

Spróbujmy teraz prześledzić jak zastosowano podane przed miesiącem zalecenia do przeróbki „z klawiatury na joystick” bardzo sympatycznego programu POOL, napisanego przez Mike'a Lamba (1983 rok).

Składa się on z dwóch segmentów:
— pierwszy (będący równocześnie programem ładującym) zawiera fragment w kodzie maszynowym (linie 1 - 130) oraz w Basicu (linie 140 - 1140) i ma długość 6975 B,
— drugi (bajty bez nagłówka) o długości 1536 B ładowany jest od komórki o adresie 30976 za pomocą procedury znajdującej się w pierwszym fragmencie pod adresem #65A7.

Z danych tych obliczamy, że cały program POOL będzie ładowany do pamięci RAM od adresu 23755 do 32512.

Przyjmujemy następujące założenia dla podprogramu obsługi joysticka:
— działanie klawiszy „s” i „a” (obrót celownika) będziemy symulować ruchem drążka „w prawo” i „w lewo”,
— funkcję klawisza „I” (siła uderzenia) — ruchem drążka „do przodu”, a klawisza „ENTER” (strzał) — naciśnięciem FIRE.

Pozostałe klawisze ("1", „2” oraz „ENTER” po „GAME OVER” w opcji 2) używane są sporadycznie podczas gry, dlatego pozostawimy ich funkcje bez zmian.

Pierwszy segment wgrany instrukcją „LOAD” uruchamia się po naciśnięciu BREAK. Obserwując „dziwaczne efekty” jego działania, należy zwrócić uwagę na możliwość ustawiania celownika (efekt dźwiękowy) oraz wykonania uderzenia. Stąd nasuwa się wniosek, że funkcji odczytu stanu klawiszy należy szukać właśnie w tej części programu. Nie ma ich w liniach Basica (znajdziemy tam natomiast fragment dotyczący klawiszy, których działania nie zmieniamy). A więc pozostaną do zbadania linie 1 do 130, napisane w kodzie maszynowym. Rzeczywiście, przy pomocy MONS'a i rozkazu G (G: DB: FE oraz G: FD: 7E: CE) odnajdujemy interesujący nas fragment programu:

```
# 6460 LD A, # FD        odczyt stanu
    IN A, (# FE)         rzędu klawiszy A—G
    LD, HL, (# 7A0E)
    RRA                  przesuwa bit 0 (stan „a")
                         do znacznika przeniesienia
    JR C, # 646D         skok, gdy nie naciśnięto „a"
    DEC HL
    JR#64E7              skok do obsługi „a"
# 646D RRA               przesuwa bit 1 (stan „s") do znacznika przeniesienia
    JR NC, #64E6         skok do obsługi „s"
# 6470 LD A, (IY-50)  wczytuje do akumulatora kod ASCII naciśniętego klawisza
    CP # 6C              sprawdza, czy to kod klawisza „1"
    El
    JR NZ, # 64BA        skok, gdy nie naciśnięto „1"
# 64BA CP # OD           sprawdza, czy to kod klawisza „ENTER"
    JR NZ, # 64DE        skok, gdy nie naciśnięto ENTER
```

Zauważymy jeszcze raz, że w programie odczyt stanu klawiatury następuje albo przez sprawdzenie bitu (klawisze „a” i „s”), albo kodu ASCII (klawisze „1” i „ENTER”). Musimy to uwzględnić podczas układania podprogramu obsługi joysticka.

Powyżej obliczyliśmy, że oryginalny program POOL kończy się w komórce 32512, a więc ulokujemy nasz podprogram pod adresem 33000 (#80E8):

```
ORG 33000           ruch drążkiem „w prawo", "w lewo"
XOR A               odczyt stanu portu joysticka
IN A, (# 15)BIT     0, A sprawdza, czy ruch „w prawo"
JR Z, E1            jeśli nie, to skok do etykiety E1
LD A, # BDł         aduje do akumulatora 189 (binarnie 10111101), symulując naciśnięcie „s"
RET                 powrót do programu

E1: BIT 1           A sprawdza, czy ruch drążkiem „w lewo" i jeśli tak,
JR Z, E2            to symuluje naciśnięcie
A, # EB             klawisza „a"
RET

E2:LD A, # FD       gdy nie użyto joysticka, podprogram odczytuje
IN A, (# FE)        stan klawiszy „a" i „s"
RET                 i wraca do programu głównego
ORG 33050           ruch drążkiem „do przodu" i FIRE
XOR A
IN A, (#15)
BIT 3, A            sprawdza, czy ruch drążkiem "do przodu"
JRZ, F1
PUSH BC
LD BC, 5000         pętla opóźniająca wykonanie procedury obsługującej klawisz „1"
A1: DEC BC
LD A, B
OR C
JR NZ, A1
POP BC
LD A, # 6C          symuluje naciśnięcie klawisza „1" (kod ASCII równy 108)
RET
F1: BIT 4, A        sprawdza czy naciśnięto FiRE
JR Z, F2            i jeśli tak, to symuluje
LD A, # 0D          naciśnięcie „ENTER" (kod ASCII równy 13)
RET
F2: LD A, (IY-50)   gdy nie użyto joysticka, podprogram odczytuje stan klawiszy „1" i „ENTER" i wraca do programu głównego
RET
```

Podprogram ten najlepiej napisać przy pomocy programu GENS 3M i nagrać go na taśmę instrukcją SAVE „POOL” CODE 33000,85.

Musimy jeszcze wprowadzić zmiany do programu głównego (do pierwszego segmentu), polegające na wpisaniu rozkazów CALL w miejsce rozkazów odczytu stanu klawiatury. Za pomocą MONS'a wpisujemy do komórek 6460 + 6462 rozkaz CALL 33000 (#CD, #E8, #80), a do komórki 6463, która pozostaje po zmianie rozkazu IN A, (#FE), rozkaz NOP (#00). Podobnie do komórek 6470 + 6472 wpisujemy CALL 33050 (#CD, #1A, #80). Poprawiony segment wgrywamy na taśmę instrukcją SAVE „POOL” LINE 170.

Teraz należy ułożyć własny program ładujący np.: 10 LOAD""CODE: LOAD"" (może on ewentualnie zawierać opis funkcji joysticka i inne uwagi), nagrać go na taśmę i w końcu, za pomocą programu COPY-COPY, zmontować poprawioną wersję gry POOL, złożoną z: naszego programu ładującego, segmentu obsługi joysticka, poprawionego pierwszego segmentu i bajtów bez nagłówka.

Powyższe zmiany możemy wprowadzać także z poziomu Basica. W tym celu, za pomocą programu COPY- COPY, wpisujemy w pierwszym segmencie następujące POKE'i:

```
25254, 205      25270, 205
25255,232       25271,26
25256,128       25272,129
25257, 0
```

oraz dopisujemy własny program ładujący:

```
10 LET s=0: FOR i=33000 TO 33021: READ a: POKE i, a:LETs = s + a: NEXT i
20 FOR i = 33050 TO 33080: READ a: POKE i, a: LET s=s + a: NEXT i
30 IF s<>6561 THEN PRINT „POPRAW DANE": STOP 40 LOAD""
50 DATA 175, 219, 21, 203, 71, 40, 3, 62, 189, 201, 203
60 DATA 79, 40, 3, 62, 190, 201, 62, 253, 219, 254, 201
70 DATA 175, 219, 21, 203, 95, 40, 13, 197, 1, 0
80 DATA 80, 11, 120, 177, 32, 251, 193, 62, 108, 201
90 DATA 203, 103, 40, 3, 62, 13, 201, 253, 126, 206, 201
```

Wgrywamy poprawiony program POOL i jeśli nie popełniliśmy żadnego błędu, możemy rozegrać partyjkę bilardu przy pomocy joysticka. Zauważymy, że nasze poprawki nie tylko ograniczyły konieczność korzystania z klawiatury. Dzięki przeniesieniu obsługi klawiszy „s”, „a” i „1” na wychylenie drążka, a strzału na FIRE (realizacja poprzez naciśnięcie przycisku), eliminujemy możliwość popełnienia fatalnego w skutkach błędu, polegającego na naciśnięciu klawisza „ENTER” zamiast „1” przy rozgrywaniu gry z klawiatury.

Na zakończenie podaję poprawki do kilku innych programów gier, które zdobyły sobie dużą popularność. Zachęcam równocześnie do ich przestudiowania wraz z oryginalnymi programami gier, w celu „przećwiczenia” innych przykładów przeniesienia sterowania z klawiatury na joystick.

JUMPING JACK

1. Układamy własny program ładujący:

```
10 LET s = 0: FOR i = 33000 TO 33022: READ a: POKE i,a: LET s=s+a: NEXT i
20 FOR i=33050 TO 33065: READ a: POKE i, a: LET s=s+a: NEXT i
30 IF s=4698 THEN PRINT „POPRAW DANE": STOP 40 LOAD""
50 DATA 175, 219, 21, 203, 71, 40, 3, 62, 30, 201, 203, 79
60 DATA 40, 3, 62, 29, 201, 1, 254, 127. 237, 120, 201
70 DATA 175, 219, 21, 203, 95. 40, 3, 62, 30, 201
80 DATA 1, 254, 254, 237, 120, 201 2.
```

POKE'i do segmentu (bajty bez nagłówka) o długości 8818 B wprowadzamy przy pomocy programu COPY-COPY:

```
29700,205      29714,205
29701,26       29715,232
29702,129      29716,128
29703,0        29717,0
29704,0        29718,0
```

SKIING

Zamiast dwóch pierwszych segmentów w Basicu wgrywamy własny program ładujący:

```
10 CLEAR 24575: LET s=0: FOR i = 33000 TO 33034: READ a: POKE i, a: LET s = s+a: NEXT i
20 IF s <> 3674 THEN PRINT/POPRAW DANE" STOP
30 LOAD"" SCREEN$: LOAD"" CODE
40 FOR i=1 TO 4: READ a: POKE a, 205: POKE a+1, 232: POKEa + 2, 128: NEXT i
50 RANDOMIZE USR 24576
60 DATA 175, 219, 21, 203, 71, 40, 3, 62, 23, 201
70 DATA 203, 79, 40, 3, 62, 15, 201, 203, 87, 40, 3, 62, 27, 201
80 DATA 203, 95, 40, 3, 62, 29, 201, 123, 219, 254, 201
90 DATA 28393, 28423, 28432, 28471
```

GULPMAN

Do programu ładującego dopisujemy następujące linie:

```
35 GOSUB 80
80 LET s=0: FOR i = 33000 TO 33041: READ a: POKE i, a: LETs = s + a: NEXT i
90 IF s <> 4029 THEN PRINT „POPRAW DANE":STOP
100 POKE 25932,205: POKE 25933,232: POKE 25934,128
110 RETURN
120 DATA 175, 219, 21, 203, 103, 40, 3, 62, 57, 201, 203, 71, 40, 3, 62, 56
130 DATA 201, 203, 79, 40, 3, 62, 53, 201, 203, 87, 40, 3, 62, 54, 201
140 DATA 203, 95, 40, 3, 62, 55, 201, 58, 8, 92, 201
```

CHEQUERED FLAG

1. Program ładujący grę o długości 28 B poprawiamy następująco:

```
10 LET s=0: FOR i = 64000 TO 64087: READ a: POKE i, a: LETs = s + a: NEXT i 20 IF s <> 8716 THEN PRINT „POPRAW DANE": STOP 30 LOAD""
40 DATA 245, 62, 253, 184, 32, 43, 175, 219, 21, 203, 103
50 DATA 40, 18, 203, 71, 40, 5, 241, 62, 23, 24, 64
60 DATA 203, 79, 40, 57, 241, 62, 30, 24, 55, 203, 71
70 DATA 40, 5, 241, 62, 27, 24, 46, 203, 79, 40, 39
80 DATA 241. 62, 29, 24, 37, 62, 239, 184, 32, 12, 175
90  DATA 219, 21, 203, 95, 40, 22, 241, 62, 30, 24, 20
100 DATA 62, 223, 184, 32, 12, 175, 219, 21, 203, 87, 40
110 DATA5, 241,62, 30, 24,3,241,237, 120, 18, 201
```

2. POKE'i do czwartego segmentu o długości 38400 B wprowadzamy przy pomocy programu COPY-COPY:

```
54545,205      54546,0      54547, 250
```

Uważny Czytelnik z pewnością dostrzeże, że w proponowanym sposobie przejścia „z klawiatury na joystick”, ładowany w podprogramie kod klawisza jest ponownie sprawdzony przez program główny. Jednakże uniwersalność tak układanych podprogramów obsługi joysticka kompensuje tę usterkę, a w praktyce nie wpływa na szybkość działania programu.

Adam Gawłowski



WŁASNE ZNAKI NA EKRANIE JOYCE'A

Komenda SYMBOL dostępna w Basicu Amstrada CPC 6128 pozwala definiować własne znaki na ekranie. Jest to niezwykle użyteczne jeśli chcemy korzystać np. z polskich liter lub tworzyć potrzebne w grach symbole.

Amstrad PCW 8256 (Joyce) nie ma wbudowanego interpretera języka Basic, a dostępny na dyskietce Mallard Basic, z powodu swej przenaszalności, pozbawiony jest procedur zależnych od sprzętu — np. dostępu do generatora znaków. Wprowadzenie nowych symboli do generatora możliwe jest poprzez dopisanie fragmentu kodu rozszerzającego system operacyjny komputera.

Rozszerzenie to nazywa RSX'em (ang. Resident System eXtension) napisano w assemblerze procesora Z80 (patrz rys. 1. Zbiór SYMBOL.MAC). Gotowy program instalujący dostęp do generatora znaków można uzyskać wykonując następujące operacje.

A>m80 =symbol

assemblacja zbioru SYMBOL.MAC przy pomocy assemblera M80.

A>link symbolfop]

konsolidacja zbioru SYMBOL.REL, utworzonego przez assembler

A>ren symbol.rsx = symbol.prl

zmiana nazwy zbioru SYMBOL.REL, otrzymanego przez program LINK z opcją OP

A>gencom symbol [null]

Utworzenie zbioru SYMBOL.COM, który należy wywołać bezpośrednio przed programem odwołującym się do generatora znaków.

Programy GENCOM.COM i LINK.COM znajdują się na trzeciej stronie dyskietki systemowej. Problem komunikacji między RSX'em a programem napisanym przez użytkownika rozwiązano w następujący sposób:

1. inicjalizacja rejestrów procesora Z80:
   C = 79, numer RSX'a
   DE = adres wektora złożonego z 9 bajtów. Pierwszy bajt wektora zawiera kod ASCII zmienianego znaku. Osiem następnych bajtów definiuje osiem kolejnych wierszy znaku.
2. Wywołanie funkcji BDOS'a (skok do adresu 0005) w Turbo-Pascalu, dzięki procedurom BDOS i ADOR, jest to wyjątkowo proste i nie wymaga żadnego fragmentu w kodzie maszynowym.

Na rysunku 2 (zbiór SYMBOL.PAS) przedstawiono program zmieniający trzy znaki o kodach 240, 241, 242. Zastosowana w programie procedura SYMBOL stanowi ogniwo pośrednie między użytkownikiem a RSX'em. Procedura ta ma 9 argumentów typu BYTE, podstawianych na wspomniany wcześniej wektor, którego adres przekazywany jest przez rejestr DE.

Uruchomienie programu pascalowego wymaga następujących kroków:

1. kompilacji na dysk zbioru SYBMOL.PAS. W wyniku tej operacji otrzymujemy zbiór SMBOL.COM. W tym miejscu tracimy zbiór SYMBOL.COM utworzony po operacji gencom symbol [null],

2. wprowadzenie komendy:

A>gencom symbol symbol

Powstały w wyniku tych operacji zbiór SYMBOL. COM można uruchomić:

A>symbol

Na rys. 3 przedstawiono rezultat działania przykładowego programu.

Uwagi

1. Ponieważ RSX zajmuje pewną część pamięci dostępnej dla użytkownika (tzw. TPA — Transient Program Area), końcowy adres zbioru typu COM, generowanego przez kompilator Turbo-Pascala, musi być mniejszy niz zwykle — np. C000J H. Adres ten ustawiany jest przez opcję E (END ADDRESS), po wybraniu 0 (compiler OPTION) kompilacji na dysk,
2. Przy wykonywaniu operacji
   A>gencon symbol symbol
   na dyskietce w stacji A: muszą się znajdować następujące zbiory:
   a. GENCON.COM
   b. SYMBOL.COM — wynikowy zbiór kompilatora Pascala
   c. SYMBOL.RSX — zbiór uzyskany ze zbioru SYMBOL, MAC po opisanych wcześniej krokach.
3. Jeśli program użytkownika, odwołujący się do generatora znaków nosi inną nazwę np. DEM0. COM, to operacja:

A>gencom demo symbol

dołączy do zbioru DEM0.COM RSX'a a SYMBOL. RSX. Uruchomienie programu DEMO przed ostatnio opisaną operacją nie da pożądanego efektu zmiany generatora znaków ekranu, ale nie zawiesi też komputera,

4. Innym sposobem uruchomienia DEM0.COM otrzymanego bezpośrednio po kompilacji jest wykonanie następującej sekwencji poleceń:

A>gencom symbol.rsx [null]
A>symbol
A>demo

W tym przypadku mamy dwa zbiory typu COM; SYMBOL i DEMO, Zbiór SYMBOL.COM instaluje RSX'a, a program DEMO korzysta z niego,

5. Ciekawy (i wygodny) efekt daje operacja

A>gencom turbo.com symbol.rsx

W jej wyniku dostajemy nową wersję kompilatora Turbo-Pascala, która pozwala korzystać z procedury pascalowej SYMBOL w trakcie kompilacji do pamięci, bez opuszczania kompilatora. Polecenie:

A>gencom turbo

odtwarza nam oryginalny zbiór TURBO, COM.

Osobom korzystającym z Mallard Basica, pozostaje niewątpliwa przyjemność napisania fragmentu programu realizującego procedurę SYMBOL przy pomocy wstawki w kodzie maszynowym

Jonasz Mayer

```
; rsx do procedury SYMBOL
;   w Turbo-Pascalu
; (C) JM Grudzien 1987
        .z80
        .cseq
; header
        ds  6
        jp  start
next:   db  0c3H,0,0,0FFH,0
        db  'symbol  '
        db  0,0,0
start:
        ld  a,c
        cp  79
        jp  nz,next
begin:
        ex  de,hl
        ld  bc,9
        ldir
        ld  hl,(0001)
        ld  bc,87
        add hl,bc
        ld  bc,code
        call   jpnl
        dw  00e9h
        ret
code:
        ld  b,8
        ld  hl,0b900H
        ld  a,(buffer)
        ld  e,a
loop:   add hl,de
        djnz   loop
        exi de,hl
        ld  hl,char
        ld  bc,8
        ldir
        ret
jphl:   jp  (hl)
        db  0
char:   ds  8
        end

  wydruk zbioru:  SYMBOL.MAC
```

```
program Symbol Demo;
(*******************************************************)
(*    Przyklad zastosowania procedury SYMBOL w programie *)
(*    pascalowym. Wykorzystano RSX'a odwolujacego sie    *)
(*    do systemu operacyjnego przez pascalowa procedure  *)
(*    BOOS.                                              *)
(*           (C) JH Grudzien 1987                        *)
(*******************************************************)

procedure symbol(CharNo, r1,r2,r3,r4,r5,r6,r7,r8 : byte);
var b : array(,0,,8,) of byte;
begin
 b(.0.) := CharNo;    b(.1.) := r1;      b(.2.) := r2;
 b(.3.) := r3;        b(.4.) := r4;      b(.5.) := r5;
 b(.5.) := r6;        b(.7.) := r7;      b(.8.) := r8;
 Bdos(79,addr(b));
end; (* of symbol *)

begin   (* SYMBOL DEMO *)
  writeln('Znaki o kodach: 240,241,242 przed wywolaniem procedury SYMBOL ');
  writeln(chr(240);10,chr(241);10,chr(242);10);
  symbol(240,$3A,$3A,$12,$FE,$3A,$38,$44,$82);
  symbol(241,$60,$68,$52,$66,$FA,$F8,$88;$88); symbol(242,24,0,126,76,24,50,126,0);
  writeln('Znaki o kodach: 240,241,242 po wywolaniem procedury SYMBOL ');
  writeln(chr(240);10,chr(241);10,chr(242);10);
end;

  wydruk zbioru: SYMBOL.PAS
```

```
M>symbol
Znaki o kodach: 240,241,242  przed wywolaniem procedury SYMBOL
                 ä         ê         ì
Znaki o kodach: 240,241,242  po wywolaniu procedury SYMBOL
                 ż         ą         ź
```

KATALOG DYSKOWY

W ostatnim numerze z 1987 roku przedstawiliśmy Wam program umożliwiający korzystanie ze spisu zawartości kasety magnetofonowej wewnątrz programu w BASIC-u. Teraz kolei na dyskietkę.

Zamieszczony poniżej listing należy uważnie wpisać do komputera i na wszelki wypadek zapamiętać na dyskietce. Rozkazem RUN uruchamiany program i odczekujemy kilkanaście sekund, aż pojawi się znany dobrze komunikat „Ready". Jeśli otrzymamy wcześniej komunikat "Blad w DATA!!!", to znaczy, że popełniliśmy błąd przy przepisywaniu i daną linię należy poprawić. Komunikaty Basica (np. "Syntax error") mogą również świadczyć o błędzie przy kopiowaniu listingu.

Dodatkową pomocą przy przepisywaniu są sumy kontrolne umieszczone w nawiasach kwadratowych na prawo od każdej linii programu. Jeżeli przed rozpoczęciem wpisywania uruchomiliśmy program „Weryfikator" z nr 1/1988, to po każdym naciśnięciu ENTER ukaże się liczba, która musi być równa sumie kontrolnej. Nierówność oznacza pomyłkę w ostatnio wprowadzonej linii.

Po pojawieniu się komunikatu „Ready" możemy skasować program komendą NEW, gdyż spełnił on swoje zadanie: procedura w języku maszynowym została załadowana do pamięci począwszy od adresu &9D00. Ponieważ ładowanie programu do pamięci z linii DATA trwa aż 15 sekund, to opłaca się zapamiętać teraz fragment pamięci rozkazem:

SAVE „DCAT.BIN",b,&9D00,&410 aby ładować go potem trzema rozkazami:

MEMORY &9CFF:LOAD „DCAT.BIN", &9D00:CALL &9D00

zamiast za każdym razem od nowa uruchamiać program w Basicu.

Procedura dodaje do Basica dwie nowe komendy RSX: |DCAT i |INFO.

Rozkaz DCAT służy do wpisania katalogu dyskietki dc tablicy stringów. Jego wywołanie wygląda następująco:

|DCAT,@a$(0),dr

Parametr dr jest wyrażeniem całkowitym dającym wartość 0 lub 1 i oznacza stację dysków, na której przeglądany ma być katalog (0 = A, 1 = B). Opuszczenie parametru dr powoduje wybranie stacji domyślnej (ustawionej przez :A,:B lub :DRIVE).

Tablica stringowa a$ (można oczywiście użyć innej nazwy) musi posiadać co najmniej tyle elementów (stringów) o długości większej od 17 ile jest zbiorów na dyskietce. Jeśli będzie ich mniej to komputer wypisze komunikat „Table too small". Zawartość stringów jest przy tym nieistotna. Komenda zamieni miejscami elementy tablicy tak, aby właściwe elementy znalazły się na początku i wpisze do nich nazwy zbiorów z dyskietki w następującym formacie

"00:NAZWA_»TYP*#z"

00 — numer użytkownika (USER); może przyjmować wartości od 00 do 15

NAZWA — ośmioznakowa nazwa pliku (ewentualnie dopełniona spacjami) np, CIOCPM3

TYP — trzyznakowe rozszerzenie nazwy (ew. dopełnione spacjami: np. EMS, BAS, BIN, COM)

* — oznacza plik zabezpieczony przed zapisem (R/O); spacja zamiast gwiazdki oznacza zwykły plik

— oznacza plik systemowy (nie pokazywany przez CAT i DIR — nasza komenda pokazuje takie pliki); dla plików zwykłych w tym miejscu jest spacja

z — jest pojedynczym znakiem, którego kod mówi o wielkości pliku tzn. jeśli ASC(z) = 15to plik zajmuje 15 kB;

Stringi zawierające nazwy zbiorów są przez DCAT sortowane alfabetycznie (tak jak przez CAT), przy czym brany jest pod uwagę również numer użytkownika, to znaczy string „00:ZZZZ..." jest przed stringiem „01 :AAAA...". Ponieważ znak z (wielkości pliku) ma najczęściej kod mniejszy od 32, czyli jest znakiem sterującym, to nie należy wyświetlać elementów tablicy w całości, lecz następująco:

PRINT (LEFT$(a$(i),17),ASC(MID$ (a$(i),18,1))

Otrzymany w ten sposób napis od 00 do # i obok wielkość pliku w kilobajtach, „i" jest numerem elementu tablicy.

Jeżeli string, do którego wpisano informację o zbiorze ma długość większą od 18, to znaki od 19 do ostatniego pozostają niezmienione.

Ze względu na specyfikę obsługi stringów w Amstradzie nie wolno w programie inicjalizować tablicy przez przypisanie pojedynczego stringu np a$(i) = „01234567890123456789". Należy posłużyć się przypisaniem dowolnego wyrażenia stringowego np.

a$(i)=„0" +„1234567890123456789" lub krócej: a$(i) = SPACE$(18)

Jeżeli tę samą tablicę używamy do wczytania katalogu z kilku dyskietek, to należy ją za każdym razem inicjalizować, by móc stwierdzić, ile zbiorów jest na kolejnej dyskietce (jeśli będzie mniej, to w nie zainicjalizowanej tablicy pozostaną stringi z informacją o zbiorach poprzedniego katalogu).

Drugą komendą, którą daje nam program jest INFO. Wywołuje się ją następująco: |INFO,@nazwa$,@a% ,@b% ,@c%, @d%,@e%

nazwa$ — jest zmienną stringowa zawierającą nazwę zbioru, o którym informację chcemy uzyskać; nazwa nie może zawierać oznaczenia stacji („a:") ani numeru użytkownika („0:"); wszystkie spacje są ignorowane; użytkownicy CPC 664 i 6128 mogą zamiast @ nazwa$ podać nazwał lub bezpośrednio string a%,b%,c%,d%,e% są zmiennymi całkowitymi o dowolnej, ale określonej wartości. Po wykonaniu komendy zostaną w nich umieszczone następujące informacje:

a% — typ zbioru:

0 — Basic

1 — Basic zabezpieczony (SAVE,p)

2 — program w języku maszynowym

22 — zbiór ASCII lub program COM

b% — adres ładowania programu maszynowego; dla Basica zawsze 368; dla ASCII nieokreślone

c% — długość zbioru; dla ASCII nieokreślone

d% — jeżeli wielkość zbioru przekracza 64 kB tutaj umieszczany jest najbardziej znaczący bajt długości zbioru (prawie zawsze 0; dla ASCII nieokreślony)

e% — adres autostartu dla programów maszynowych uruchamianych komendą RUN; dla Basica — 0; dla ASCII nieokreślone

Nazwy zmiennych mogą być oczywiście różne od tych, podanych w przykładzie.

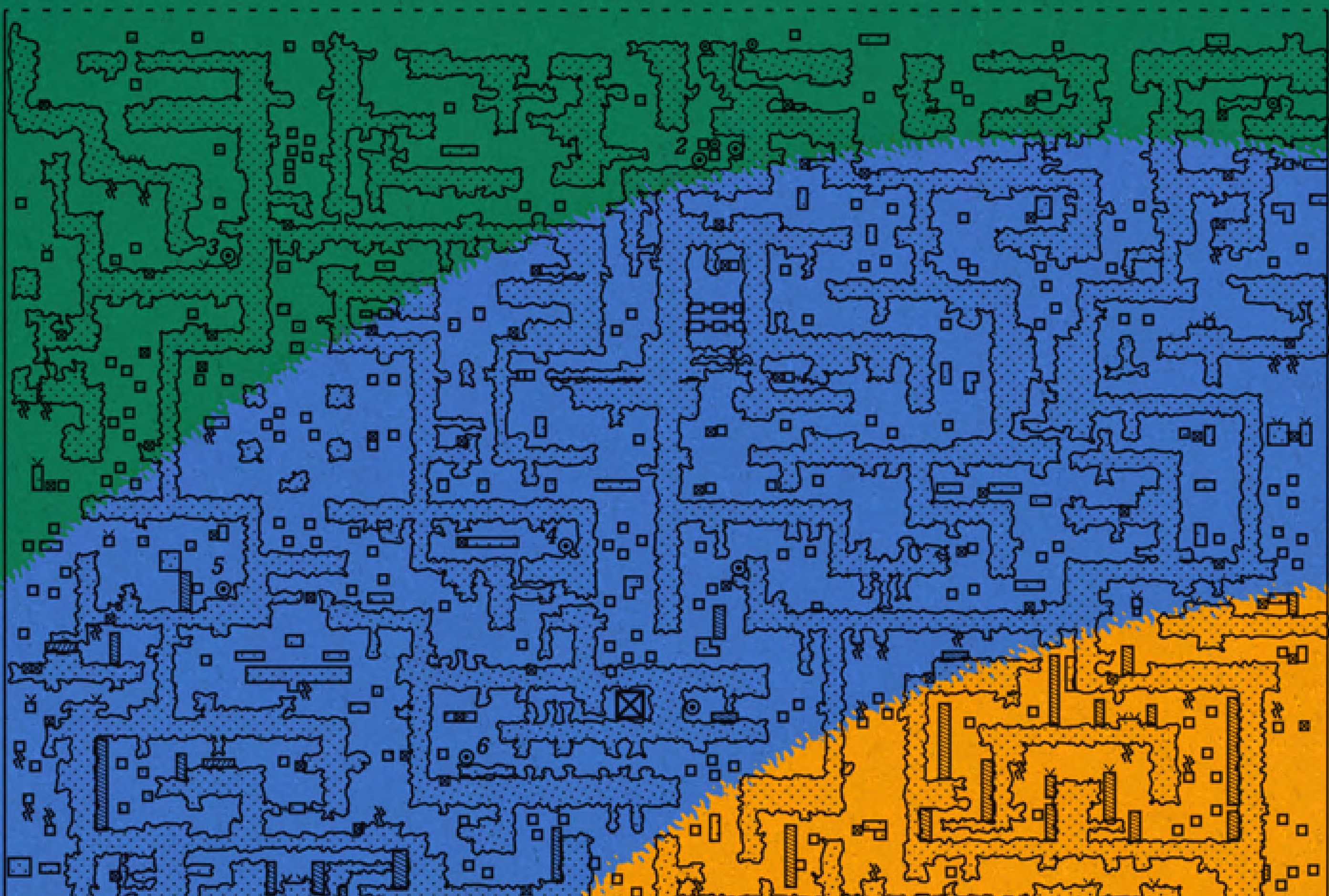
Zbiór szukany jest zawsze na domyślnej stacji dysków i domyślnym obszarze użytkownika (które można ustawić komendami :A, :B, :DRIVE i :USER). Jeżeli nie zostanie znaleziony, to komputer wypisze: „File not found", lecz nie przerwie programu.

Przedstawiony w numerze 12/1987 katalog kasetowy można załadować do pamięci razem z katalogiem dyskowym, co umożliwia na przykład pisanie programów automatycznego kopiowania dyskietek na kasety i odwrotnie. Najpierw należy załadować poprzedni program a dopiero potem przedstawiony powyżej..

Sergiusz Wolicki

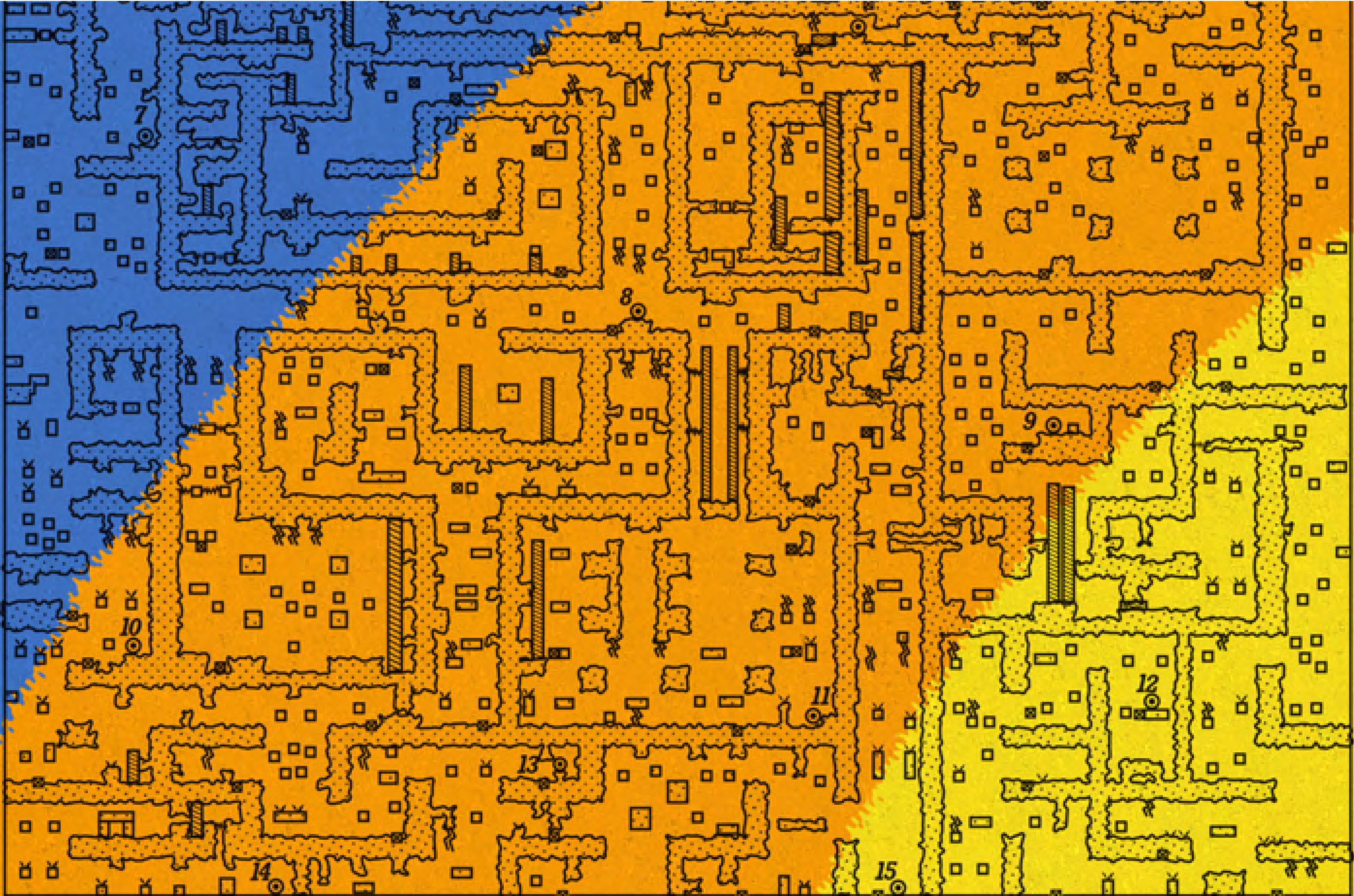
```
10 DATA 21119d01099dc3d1bc159d   [1218]
20 DATA c3569ec31e9d3535303044   [1282]
30 DATA 4341d4494e46cf00210000   [1847]
40 DATA 39220fa1f52a7dbe7e321b   [1976]
50 DATA a1237e321aa1f1fe06c251   [18151
60 DATA 9f0605dde5cdb39f2b7e3d   [1413]
70 DATA c2519f10f5cdb39f7eb7ca   [1808]
80 DATA 519f47dde1237e23666f11   [1699]
90 DATA 29a13a1aa112130e097ecd   [1593]
100 DATA ed9f23fe20280ffe3aca51  [13793
110 DATA 9ffe2e28120dca519f1213  [1213]
120 DATA 10e6cdbe9f0e04cdbe9f18  [1251]
130 DATA 23cdbe9f0e040528f37ecd   [150]
140 DATA ed9f23fe2028f4fe3aca51  [1662]
150 DATA 9ffe2eca519f0dca519f12   [513]
160 DATA 1318e23e001213121312dd  [1208]
170 DATA e5cd81a0cd62a0d2609fe5   [949]
180 DATA cdc69fe120f31110001916  [1272]
190 DATA 007eb7175f3001143a1ca1   [902]
200 DATA e6c02109002003210800eb  [1079]
210 DATA 0600ed520430fb19053a1c  [1248]
220 DATA a1e6c0b53c321ca13a1da1  [2243]
230 DATA 80321da1cdd6a00643213b  [1454]
240 DATA a1fd21000016005e23fd19  [1080]
250 DATA 10fa5e2356fde5e1fd213b   [623]
260 DATA a1b7ed520e162003fd4e12  [1608]
270 DATA dde1cdb39ffd7e1a7723fd  [1120]
280 DATA 7e1b77cdb39ffd7e427723  [1883]
290 DATA af77cdb39ffd7e407723fd  [1226]
300 DATA 7e4177cdb39ffd7e157723   [949]
310 DATA fd7e1677cdb39f7123af77  [1663]
320 DATA c921000039220fa1f52a7d  [1677]
330 DATA be7e321ba1f13d281a3dc2  [1028]
340 DATA 519fdd7e01b7c2519fdd7e  [1235]
350 DATA 00fe02d2519fdd23dd2332  [1335]
360 DATA 1ba1dd6e00dd66012211a1  [1584]
370 DATA 2213a12b2b2b7e3dc2519f   [795]
380 DATA 2b562b5e2b7efe02c2519f  [2071]
390 DATA 192217a12100002215a1cd   [857]
400 DATA 81a0cd62a0d07efe1030f7  [1421]
410 DATA cdf69fdd2a11a12a15a1eb  [1566]
420 DATA 7bb228121bd5cdfaa0d128  [2730]
430 DATA 703808dd23dd23dd2318ea   [595]
440 DATA dde52a13a1eb2a17a1b7ed  [1409]
450 DATA 52da5b9fd5dde1cdf0a0fe  [2125]
460 DATA 12300513131318e83220a1  [1426]
470 DATA 2221a1eb2129a1011200ed  [1674]
480 DATA b0dde5d1e1ebb7ed52e5c1  [1822]
490 DATA dde5e1d5e5d113132b78b1  [1615]
500 DATA 2802edb8e13a20a17723ed  [1794]
510 DATA 5b21a17323722a15a12322  [1578]
520 DATA 15a12a13a12323232213a1  [1188]
530 DATA c3af9ecdf0a0e5dde13a3a  [1807]
540 DATA a1dd8611dd7711c3af9e21  [1948]
550 DATA 719f180d21839f18082190  [1206]
560 DATA 9f180321a29fed7b0fa17e   [584]
570 DATA fe24c8cd5abb2318f65061  [1395]
580 DATA 72616d6574657220657272  [1377]
590 DATA 6f720d0a24446973632065  [1427]
600 DATA 72726f720d0a245461626c  [1084]
610 DATA 6520746f6f20736d616c6c  [1758]
620 DATA 0d0a2446696c65206e6f74  [1453]
630 DATA 20666f756e640d0a24dd6e  [1248]
640 DATA 00dd6601dd23dd23c93e20   [972]
650 DATA 0dc8121318fa1129a11abe  [1194]
660 DATA c0060b13237ee67f4f1ab9  [1417]
670 DATA c010f513237ee61f4f1ab9  [1027]
680 DATA c0131323237ee63f4f1ab9  [1282]
690 DATA c9fe61d8fe7bd0d620c911  [2165]
700 DATA 29a17e23c600274fe6f00f  [1098]
710 DATA 0f0f0fc630121379e60fc6  [1597]
720 DATA 3012133e3a121306087ee6   [720]
730 DATA 7f12231310f83e2e121301  [1656]
740 DATA 00007e2307cb101f12137e  [1342]
750 DATA 2307cb111f12137ee67f12  [1511]
760 DATA 23133e2acb3838023e2012   [469]
770 DATA 133e23cb3938023e201213   [590]
780 DATA 2323237e0f0f0f4fe6e028   [627]
790 DATA 010c79e61f1213c92a1ea1  [1218]
800 DATA 11200019221ea13a19a13c  [1582]
810 DATA 3219a1fe40d0e60f37c0cd  [1149]
820 DATA d6a0221ea1c9210ba13a1b   [751]
830 DATA a1b72803210ca1cdd4bcd2  [1255]
840 DATA 569fcd1b00210da1cdd4bc  [1579]
850 DATA d2569f2223a1793225a121  [1479]
860 DATA 0ea1cdd4bcd2569f2226a1  [1747]
870 DATA 793228a12a42be110d003a  [1104]
880 DATA 1ba1b72803114d00197e32  [1357]
890 DATA 1da123237e321ca13eff32  [1154]
900 DATA 19a1c93a1ba15f3a1da157  [1238]
910 DATA 3a1ca14f3c321ca1213ba1  [1462]
920 DATA df23a1d2569fc9dd7e00dd  [1935]
930 DATA 6e01dd6602c9cdf0a01129  [1898]
940 DATA a1060f1abe1323d8c010f8   [960]
950 DATA c9c1284850000000000000  [1601]
960 '                            [117]
970 ' KATALOG DYSKOWY            [1585]
980 '(c) S.W.                    [413]
990 '                            [117]
1000 IF HEX$(HIMEM,4)<"A33A" THEN   [7742]
     PRINT "Zaladuj jeszcze raz
     po zrobieniu RESET.":END
1010 MEMORY &9CFF:c!=0           [618]
1020 FOR k=0 TO 94               [1382]
1030 READ a$                     [309]
1040 FOR l=0 TO 10               [439]
1050 byte=VAL("&"+MID$(a$,2*l+1,2))  [816]
1060 POKE &9D00+k*11+l,byte      [1743]
1070 c!=c!+byte                  [1051]
1080 NEXT l,k                    [426]
1090 IF c!<>109750 THEN "Blad    [1972]
     w DATA !!!":END
1100 CALL &9D00                  [621]
```

STARQUAKE



Bajtek





Centrum - reaktor
Platforma
Liny i pnącza
Niebezpieczna antenka
Iskra elektryczna
Stacja teleportacji
Winda
Drzwi wymagające kart-klucza

W roku 2100 i Ziemi wystartował kosmolot na poszukiwanie innych układów planetarnych. Podczas lądowania na planecie X kosmolot uległ rozbiciu na skutek błędu pilota Bloba. W dodatku elementy stosu atomowego napędzającego pojazd rozsypały się po całej planecie. Celem gry jest zebranie wszystkich elementów i umieszczenie ich w sterowania lotem. W trakcie wędrówki po planecie napotkasz bardzo wiele utrudnień. Są tu drzwi zamknięte na zamek szyfrowy, który może otworzyć tyko komplet kart kluczowych z odpowiednimi numerami lub karta uniwersalna z literą A. Ponadto znajdziesz przejścia zamknięte na klucz, zapadnie i tajne przejście między skałami.

Blob ma tylko jedną kieszeń w której mieszczą się cztery przedmioty. Ponadto może niektóre przedmioty wymieniać na inne w specjalnych piramidkach jeśli posiada kartę uniwersalną.

Planetę X zamieszkują stworki wrogie naszemu kosmonaucie. Zetknięcie z nimi zmniejsza energię Bloba a niektóre zabijają go od razu. Utraconą siłę zregenerować możesz zbierając pakunki z energią. W pakunkach tych znaleźć możesz oprócz energii platformy lub naboje, czasem dodatkowe „życie”.

Pilot Blob porusza się w lewo lub w prawo. Może też podstawić sobie platformę, która po chwili zniknie. Na planecie znajdzie ruchome podesty na których lata we wszystkie strony lecz nie może przechodzić przez drzwi. Poruszanie po planecie ułatwiają także liczne stacje teleportacji oznaczone kodami słownymi. Centrum sterowania znajduje się w pobliżu stacji o nazwie QUAKE.

Gra ta jest bardzo bogata graficznie i dźwiękowo. Grać można joystickiem lub definiować przyciski. Dodatkowe funkcje to: góra — wejście do piramidki lub podniesienie przedmiotu i dół — położenie kolejnej platformy.

Krzysztof Białkowski

Autor: Stephen Crow
Firma: Bubble Bus
Komputer: ATARI 800 XL/130 XE, Commodore 64/128, Spectrum 48/+

STARQUAKE w wersjach na komputery Amstrad i Commodore różni się od wersji tu opisanej nazwami stacji teleportów. Oto pełen wykaz, nadesłany przez Seweryna Szczepanka z Olsztyna:

	SPECTRUM	ATARI	COMMODORE
1.	RAMIX	TRAID	SIGMA
2.	VEROX	KERNX	ASTRA
3.	TULSA	WHOLE	HYLIS
4.	ASOIC	MINIM	KAPPA
5.	DELTA	SALCO	FFMUR
6.	OUAKE	ATARI	CHASM
7.	ALGOL	PENTA	MALIS
8.	EXIAL	COSEC	METRE
9.	KYZIA	Z.A.P.	XENON
10.	ULTRA	ARTIC	COSIN
11.	IRAGE	CRASH	PLASM
12.	OKTUP	ARGON	OPTIC
13.	SONIO	QUARK	POLAR
14.	AMIGA	DELTA	Z.A.P.
15.	AMAHA	SECON	MESON



10 BAJTKOWA LISTA PRZEBOJÓW (4/88)

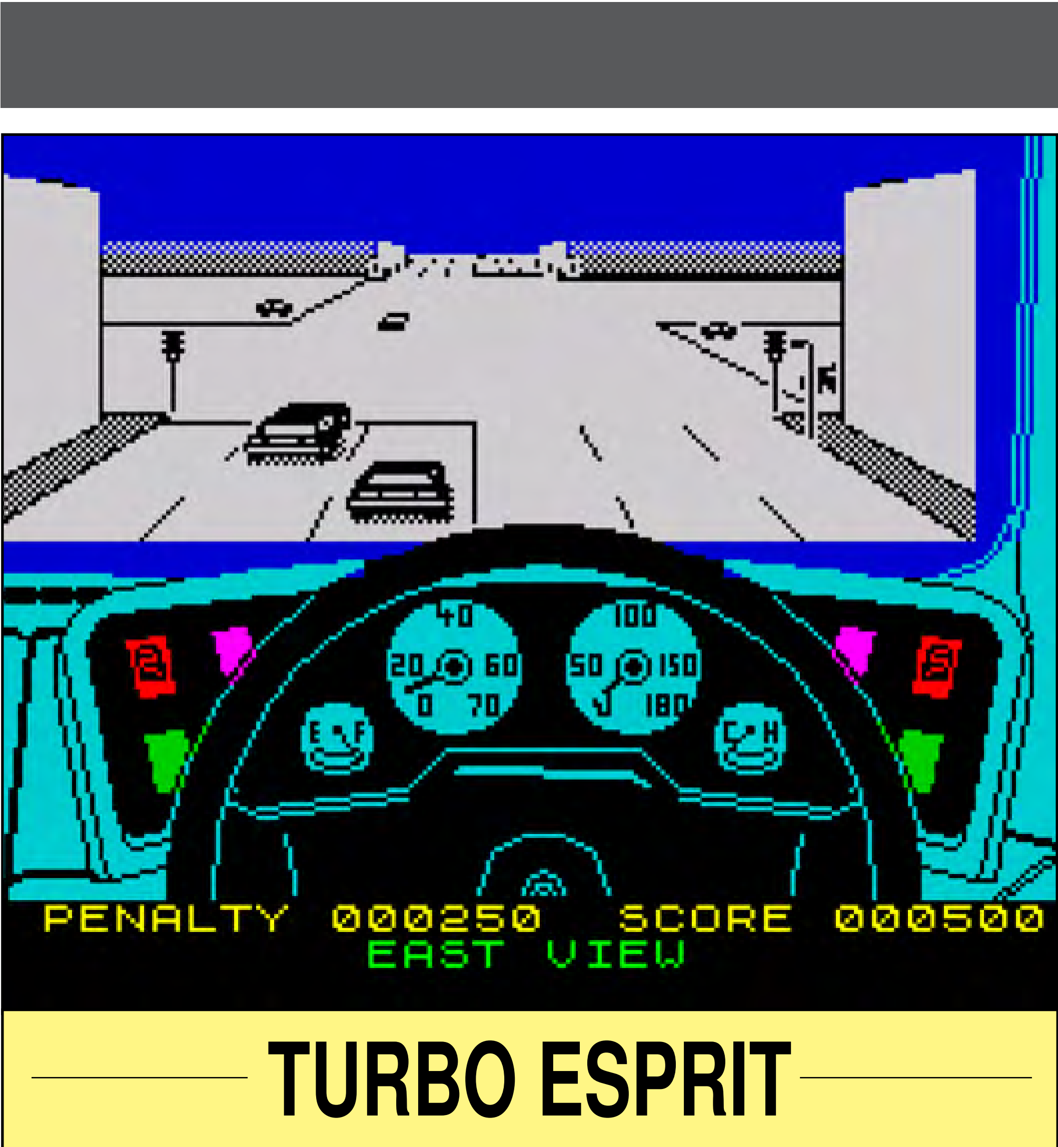
Otrzymałem już kilka opisów — bardzo dziękuję. Mam prośbę: nie opisujcie gier typu bij-zabij. Legendę można zawsze wymyśleć, a przecież chodzi o uchwycenie właściwego sensu i celu gry, wraz z ew. podpowiedziami. Na piąte notowanie nadeszło 2185 propozycji na 149 tytułów.

		ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	SABOTEUR II		x	x	x
2	ARKANOID	x	x	x	x
3	GUN FRIGHT		x		x
4	CHIMERA	x	x	x	x
5	STARQUAKE	x	x	x	x
6	TRASHMAN		x	x	x
7	EXOLON			x	x
8	EQUINOX		x		x
9	MIKIE ↓		x	x	x
10	TURBO ESPRIT !		x	x	x

Nagrody otrzymują: Gabrysia Czarnecka i Dominik Molski.



Kopertę nadesłał Bartek Antoniak z Raszyna



TURBO ESPRIT

W rozrywkowej części światowego software'u już chyba niczego nowego nie można wymyśleć. Nieomal każda nowopowstała gra jest urozmaiconą w różny sposób wersja pewnego standardu. Poza grami typu "bij-zabij!" bodajże najwcześniej wymyślono symulację jazdy samochodem. W roku 1986 firma Durell jako kolejna podjęła ten temat... (W tym miejscu wzdychasz drogi Czytelniku, machasz ręką i mamroczesz niezrozumiale coś o Pole Position, Road Race...).

Na pewno Mike Richardson przewidział to i dlatego wymyślił połączenie kilku „starych” pomysłów. Chcesz jeździć szybkim samochodem? Proszę bardzo! Lotus Esprit Essex Turbo stoi do Twojej dyspozycji (w rzeczywistości 210 koni mechanicznych i 245 km/h) — to chyba niezły pomysł, prawda?

James Bond miał w swoim Aston Martin karabin maszynowy, dlaczego Ty nie mógłbyś go mieć? Ściganymi obiektami będą samochody handlarzy narkotyków. Jak widzisz, nic nowego. Jeszcze raz możesz być stróżem prawa (i to nawet przed rozpoczęciem gry, gdyż firma w specjalnym anonsie apeluje o wskazywanie im osób kopiujących nielegalnie ich programy — 100 funtów nagrody).

Ruszamy! Najpierw formalność: wybór miejsca akcji, (jedno z czterech angielskich miast), definiowanie klawiszy, wybór stopnia trudności (od 1 do 4). Gra ma staranną oprawę dźwiękową, menu pozwala wybrać na początku jedno z czterech miast a następnie udostępnia do wglądu tabelę wyników, tabelę kar, pozwala zapisać tabelę wyników na taśmie lub załadować własną. Po zdefiniowaniu klawiszy (bardzo ważny jest FIRE) najlepiej zacząć od punktu 7, tj. od ćwiczeń w prowadzeniu pojazdu. W czasie jednej gry wolno Ci rozbić go cztery razy, ale podczas treningu jeździsz do pierwszej kraksy.

Czy gra ta jest symulatorem? Raczej nie, gdyż sama „jazda”, nie daje pełnej satysfakcji. Końcowy efekt osiągniesz jedynie wówczas, gdy przeszkodzisz przestępcom w sprzedaży narkotyków. Rozpoczynając grę, skup się więc natychmiast na pościgu. U dołu ekranu pojawiać się będą komunikaty z Centrali o zauważeniu poszukiwanych samochodów w pewnych regionach miasta (korzystaj z planu i otrzymanych współrzędnych) oraz ewentualnym zbliżaniu się któregoś z nich (dla odróżnienia są innego koloru). Ilość punktów zwiększa się po przechwyceniu narkotyków lub po zniszczeniu pojazdu przestępców. Więcej zdobędziesz za unieruchomienie go i doprowadzenie do poddania się. Karę otrzymasz za każde uchybienie w prawidłowym prowadzeniu Twego Esprita.

Na ulicach trójpasmowych jest najłatwiej, ale spróbuj skręcić przy dużej prędkości w jedno pasmową uliczkę! (Zakręca się naciskając jednocześnie FIRE i kierunek skrętu). Wystrzegaj się ulic ślepych (od tego masz plan) — bardzo trudno jest potem zawrócić. W potrzebie skorzystaj z usług stacji benzynowej (trzeba podjechać do niej poboczem i zatrzymać się przy dystrybutorach).

Po zakończeniu gry, od ilości punktów zdobytych odejmowana jest ilość punktów karnych, a różnicę zapisuje się w odpowiedniej tabeli. Jednym z poważniejszych mankamentów tej gry jest brak możliwości zmiany miejsca akcji (miasta), gdyż po zakończeniu możesz korzystać tylko z menu głównego. Drobne braki nie są jednak ważne. Turbo Esprit dla młodych kierowców! Symulacja i emocje, także możliwość rozluźnienia napiętych nerwów — autor zadbał o to, by na ulicach nie brakowało przechodniów do przejechania, ani robotników na drabinach do zrzucania ich stamtąd. Ach! Drobny szczegół... Nie zapomnij, że w Anglii obowiązuje ruch lewostronny...

Autor: Mike Richardson
Firma: Durell Software
Komputer: ZX Spectrum 48/ , Commodore 64/128, Amstrad

(pb)



SPLITTING IMAGES

Na pewno Widzisz codziennie w telewizji i gazetach twarze sławnych ludzi — polityków, przywódców, konstruktorów. Czy potrafisz odtworzyć je z pamięci? Sprawdzić to możesz wgrywając program Splitting Images napisany przez firmę Domark. Pomysł oparty jest na popularnym w angielskiej telewizji programie pod tym samym tytułem.

Po załadowaniu gry (w wersji Spectrum) na ekranie ukazuje się plansza służąca do wyboru rodzaju sterowania oraz do rozpoczęcia gry.

Cały ekran podzielony jest na dwie części: informacyjną i właściwą. Część informacyjna to zegar, licznik punktów, licznik możliwych pomyłek oraz zdjęcie układanego właśnie wizerunku. Część właściwa to kwadrat 5x5 pól. Lewe pole oznaczone strzałką służy jako wypluwacz części obrazka. Wyplucie możliwe jest po najechaniu kursorem na strzałkę i wciśnięcie FIRE. Z kwadracika wylatuje jeden lub więcej przystających kwadratów z częściami układanej postaci. Czasem zamiast niej wpada bomba z palącym się lontem, rakieta, kran, flaga lub różne inne drobiazgi. Nie wyrzucona w porę bomba wybuchnie, zabierając Ci jedno „życie". Możesz ją zalać wodą z kranu, jeśli masz go na planszy.

Po naprowadzeniu kursora na jeden z kwadratów przedstawiających część wizerunku, na bocznym zdjęciu, w części informacyjnej zobaczysz biały kwadrat. Jest to miejsce, w którym powinien znajdować się dany kwadracik. Gdy teraz naciśniesz FIRE i poruszysz joystickiem np. w lewo, część obrazka, na której jest kursor, posunie się w lewo aż do napotkania pierwszej przeszkody (ściana lub inny obrazek). Gdy na ścianie znajduje się zadrapanie, obrazek odbije się od niego w przeciwną stronę.

Gdy ułożysz już cały obrazek, zdziwisz się jak dokładna może być grafika Twojego komputera i otrzymasz do ułożenia następny wizerunek. W miarę upływu czasu gra staje się coraz trudniejsza (więcej bomb, mniej czasu, bezwładność przesuwanych kawałków). Ostatni obrazek układasz nie widząc go. Gra się tu na wyczucie, patrząc na zdjęcie w informacyjnej części ekranu.

Gra jest bardzo zajmująca; łączy w sobie elementy logiki i zręczności. Jest dopracowana pod każdym względem i na pewno spodoba się każdemu.

Firma: Domark
Komputer: Commodore 64/128, ZX Spectrum 48K/+

(mp)

KRÓL I KRÓLOWA GIER



GABRYSIA CZARNECKA lat 5
Przedszkole Nr 219 w Warszawie Zainteresowanie: gry komputerowe, bajki, książki z obrazkami. Ulubiona gra Bomb Jack Posiadany komputer Spectrum. Gabrysię nauczył korzystać z komputera starszy brat 12 letni Hubert.



DOMINIK MOLSKI lat 12
Szkoła Podstawowa nr 222. Warszawa.
Zainteresowania: sport, muzyka niepoważna, komputery.
Ulubiona gra: Silent Service.
Posiadany komputer: Atari 65XE (od grudnia 1987)

S.O.S.

Ratujcie! Poszukuję instrukcji do gry B-1 NUCLEAR BOMBER na Atari 65 XE.
Zygmunt Piętak ul. Piastowska 1 A m 4 41-100 Siemianowice

Pilnie szukam nieśmiertelności w grze KISSIN KOUSINS na Atari 65 XE.
Damian Wołczyński ul. Słowackiego 34 m 18 42-400 Zawiercie

Na czym polegają następujące gry: KNIGHT RIDER, MIAMI VICE, STEER HAWK, MUTATIONS? Mam komputer Spectrum+2.
Piotr Waśko ul. Żurawia 31 m 9 59-300 Lubin

Liczę na waszą pomoc — potrzebny mi jest opis do gry BATTLE FOR MIDWAY na Spectfum 48.
Marcin Szubski ul. Niecała 6 m 62 00-098 Warszawa

Pomóżcie mi. Potrzebuję dokładny opis SILENT SERVICE na Atari 800 XL.
Damian Smażyński ul. Poniatowskiego 19 m 11 41-500 Chorzów

Mam 11 lat i jestem posiadaczem Spectrum. Nie wiem, gdzie jest śmigłowiec w grze SABOTEUR. Pomóżcie!
Marcin Wrona ul. Noskowskiego 6 m 13 58-506 Jelenia Góra

Moja prośba dotyczy dokładnego opisu gier: CHIMERA, HOBBIT i BROADSIDES. Dwóch ostatnich nie potrafię nawet rozpocząć.
Czesław Nowak ul. Sokolska 35 44-273 Rybnik 11

Nie mogę uruchomić takich gier na C-16: LEAPER, VIDEO POKER, OPERATION HAWALL. Proszę o pomoc.
Anita Kwiatkowska ul. Tkacka 15 m 2 58-500 Jelenia Góra

Mam 16 lat. Jestem inwalidą „przykutym" na stałe do wózka inwalidzkiego. Od kilku miesięcy mam Atari 65 XE. Może znajdzie się ktoś, kto udostępni mi ciekawe gry (na kasetach) i programy na Atari. Najbardziej zależy mi na: ROAD ROCK, MONTEZUMA'S REVENGE, PHARAOH'S CURSE, PITSTOP II, PITFAL I i II.
Janusz Nawrocki ui. Prószkowska 25 m 15 45-759 Opole

Liczę na pomoc w zdobyciu opisów do gier: WINTER GAMES, MISS PACMAN, WARHAWK w wersji na Atari 65 XE.
Tomasz Słowik ul. Robotnicza 18 73-120 Chociwel

Nie wiem, jak skończyć grę MONTEZUMA'S REVENGE. Dochodzę do kaganka, zabieram go i potem krążę w kółko. Proszę o ratunek!
Magda Pająk ul. Ziębią 24 m 1 44-100 Gliwice

Mam kolegę, który ma komputer Atari 600 XL z magnetofonem, ale przychodzi z nim do mnie, gdy mam jakieś ciekawe gry. Ja do niego nie chodzę, ponieważ nie lubię poruszać się na wózku inwalidzkim. Wolę przebywać w domu. Pragnąłbym, aby ktoś przysłał mi chociaż programy gier na kartkach. Na komputer, niestety, nie stać mnie.
Michał Przybylski ul. Daleka 27 m 54 25-319 Kielce

Proszę o pomoc w grze Asterix and the Magic Cauldron w wersji na Spectrum. Gdzie jest piąty kawałek kotła i co robić dalej?
Robert Heber ul. Chorzowska 20 m 42 41-910 Bytom

Proszę o pomoc w ukończeniu gry w MONTY MOLE-2. Przechodzę przez wiele komnat oraz dwa mury, które giną po ostatnim skarbie i dochodzę do komnaty w której jeździ wózeczek do pochyłej blokady. Nie wiem jak zlikwidować mur i zebrać kolejne skarby.
Teresa Czarnecka ul. Reymonta 23 m 239 01-840 Warszawa

LISTY DO LISTY

Droga Redakcjo!

Przeczytałem list Arkadiusza Szyszko z ostatniego numeru Bajtka dotyczący gry CHUCKIE EGG. Posiadam także od niedawna komputer ATARI 130 XE i magnetofon firmowy. Widziałem wersję tej gry na SPECTRUM i moim zdaniem wersja na ATARI prawie niczym się od niej nie różni. Arek napisał, że ostatnie „życie" stracił na 21 poziomie. Ja natomiast kilka razy męczyłem się na 24 z nadzieją, że jest to ostatni poziom. Zdziwiłem się, gdy przeszedłem ten poziom rozpoczynając dalszą zabawę w pierwszym pomieszczeniu. Ale zobaczyłem nie dwie lecz pięć kur i latającą kaczkę. Przez dalsze pomieszczenie z trudem przeszedłem do 32 poziomu gdzie zakończyłem grę utratą ostatniego życia. Nie pamiętam dokładnie, ile zdobyłem punktów, ale na pewno powyżej 400 tysięcy.
Chciałbym poinformować czytelników Bajtka, co ich czeka po 21 poziomie. Do 24 poziomu, tak jak opisał to Arek, chodzą kury i lata kaczka. Na 25 poziomie znów jesteśmy w pierwszym pomieszczeniu, ale z większą ilością kur i oczywiście latającą kaczką. W dalszych pomieszczeniach będzie też więcej kur aż do 32 poziomu. A co dalej — nie wiem. Do 32 poziomu doszedłem w ponad godzinę i 3'0 minut. Premię w postaci dodatkowego „życia" otrzymujemy po przejściu każdej komnaty.

Robert Lipski
ul. Niepodległości 6/16
20-245 Lublin

WYCIECZKA W PRZYSZŁOŚĆ

KORESPONDENCJA WŁASNA

Biegnąca przez Hanower autostrada na tydzień staje się drogą jednokierunkową. Owa jednokierunkowość polega jednak na tym, że gdy rano wszystkie samochody mkną tylko w kierunku „do terenów targowych", w południe policjanci zmieniają znaki i wszystkie sześć pasów ruchu skierowanych jest „od".

Podobnie jak ruch samchodów, również wszystkie inne przejawy życia miasta są podporządkowane jednemu: wszystko dla CeBITu i jego gości. Bo CeBIT każdemu fanowi informatyki i każdemu biznesmenowi działającemu w branży komputerowej powinien wydać się rajem na Ziemi.

O przepustkę do tego raju „Bajtek" zabiegał dość długo. Dopiero jednak w tym roku udało nam się dopiąć celu i nasza „ekipa" znalazła się pośród 440 tys. szczęśliwców, którzy pomiędzy 16 a 23 marca wkroczyli do liczących w sumie 300 tys. metrów kwadratowych powierzchni hal targowych. 2300 wystawców z 40 krajów zaprezentowało w nich wszystko, co tylko kojarzy się, bądź czasem nie bardzo, z komputerami i ich techniką.

Z tym kojarzeniem rzeczywiście mogą być kłopoty. Cóż jednak robić w sytuacji, gdy de facto komputerem jest zarówno kasa sklepowa, część linii produkcyjnej maszyna do pisania, telefon jak i aparat fotograficzny?

Stoiska największych firm mogłyby z powodzeniem stanowić samodzielne wystawy. Ich obsługa liczyła czasem i kilkaset osób - hostess, specjalistów od oprogramowania, handlowców. Jak żartowali niektórzy Epson czy Nixdorf mieli stale przy swoich eksponatach więcej obsługi niż zwiedzających.

Ci ostatni nie zawiedli. Przez cały dzień od dziewiątej rano do szóstej wieczorem we wszystkich salach kłębiły się tłumy. Średnia wieku zwiedzających podobna była do tej, jaką obserwowaliśmy na warszawskiej wystawie Komputer 88 — ok. 15—17 lat Umiejętności — zróżnicowane. W przeciwieństwie jednak do zwyczajów panujących na rodzimych wystawach, wszystkiego niemal wolno było dotknąć, spróbować. Najwięcej frajdy mieli młodzi goście wystawy w „Obozie komputerowym" zlokalizowanym w jednej z hal. Tam właśnie można było sprawdzić swe umiejętności, podyskutować o sprzęcie i programach, a nawet otrzymać ciekawą ofertę pracy w zawodzie związanym z informatyką.

Dość jednak o atmosferze i widzach. Co zaprezentowały firmy znane z łamów „Bajtka"? — to chyba pytanie nr 1 każdego komputerfana. Otóż generalnie trzeba powiedzieć, że żadnych rewelacji w dziedzi nie sprzętu „powszechnego użytku" nie zaprezentowano, chyba głównie z tego powodu, że rewelacje te powstały już wcześniej. Mowa oczywiście o Commodore Amiga (na CeBIT obecne były wersje 500, 1000, 1500, i 2000) oraz Atari serii ST(520, 1040, Mega, Mega ST2). Amiga, chociaż droższa, m.in. dzięki swej wspaniałej grafice sprzedaje się znakomicie i jak powiedział nam w wywiadzie (pełny jego tekst znajdziecie w mającym się ukazać wkrótce specjalnym wydaniu „Bajtka" „Tylko o Commodore") dyrektor generalny Commodore Electronics Ltd. Paul Moloney koncern nie nadąża z produkcją za rosnącym lawinowo popytem na ten komputer. Paul Moloney przekazał nam również sensacyjne wręcz wieści na temat planowanej przez Commodore ekspansji na rynek polski (szczegóły we wspomnianym wywiadzie!).

Commodore wyróżniał się spośród innych wystawców jeszcze jednym. Jako jedyna firma dumnie i z podniesionym czołem lansował wciąż 8-bitowy mikrokomputer — „nieśmiertelny" C-64 w czwartym już wydaniu. Atari pokazało 65XE już tylko jako element gry telewizyjnej. Amstrad nie zarzucił wprawdzie produkcji modelu 6128 lecz go nie eksponował, zaś dotychczasowy partner Amstrada na rynku zachodnio- niemieckim — Schneider przestawił się już wyłącznie na PC.

Perfekcja Amigi, Atari ST, a w dziedzinie 8-bitowców C-64 tak chyba przytłoczyła potencjalnych konkurentów, że w zasadzie na wymienionych przez nas komputerach kończy się przedstawiona na CeBIT lista maszyn domowych (jeśli za taką uznać można w ogóle np. Ami-



Najnowsza popularna drukarka "Stara" — LC-10 Colour miała na CeBIT swoją światową premierę







Rodzinę PC reprezentowały zarówno potężnie wyglądające wieże, jak i portable.

gę 1000). Setki mniejszych producentów rzuciły się głównie w kierunku linii PC tam szukając handlowej szansy.

”Klony” IBM lub dokładniej mówiąc, komputery stosujące te same systemy operacyjne i akceptujące większą lub mniejszą część oprogramowania IBM były w ofercie niemal wszystkich firm, nawet takich nie kojarzących się nam dotąd z tego typu produkcją jak: Ricoh, Canon, Epson czy Rank Xerox. Jednak największą „siłą ognia” dysponowały w tej specjalności firmy dalekowschodnie. Były ich na CeBIT setki. Niektóre dopiero wchodziły na europejski rynek, szukały tu dystrybutorów i dealerów. Pokazały m.in. pierwsze kopie IBM PS2. Jak zapewnił nas np. pan William Wang z firmy Pro Systems oferowany przez niego SE-30 to dokładny odpowiednik PS2-30, tyle że o jedną czwartą tańszy. A jeszcze niedawno IBM mówił o „niekopiowalności” swego nowego produktu!

Prawdziwa rewolucja odbywa się w dziedzinie przenośnych PC. Najróżniejsze portable Toshiby, Sharpa, Amstrada i setek innych firm mają dziś niewyobrażalne wręcz: możliwości. Oparte na procesorach Intela 80286 i 80386 wyposażone są w znakomite plazmowe i ciekłokrystaliczne ekrany, 1 MB pamięci operacyjnej, często twarde dyski, a także (w miejsce jednego z 3,5 calowych floppow) w stacje 550-megabajtowycń optycznych dysków CD-ROM.

W ogóle możliwości rozbudowy pamięci w komputerach wzrastają błyskawicznie. Mimo wzrostu cen kości RAM dotychczas stosowanych pojemności (!) już wprowadza się nowe — 4 MB. W dziedzinie pamięci optycznych najciekawsze chyba były wspomniane CD i taśma optyczna zaprezentowana przez ICI Electronics o pojemności 300 gigabajtów z możliwością rozszerzenia do 300000 w przypadku polepszenia jakości laserów służących do zapisu i odczytu.

Wśród producentów drukarek trwa wciąż niezwykle ostra rywalizacja bez wyraźnych faworytów. Posiadacz największego chyba stoiska — Epson zaprezentował drukarkę 48-igłową. Za dwa lata pokażą 256-igłową — podkpiwali konkurenci. Rzeczywiście pomysł, chociaż ciekawy technicznie, (jakość druku zadziwiająca), nie przyjmie się chyba choćby z tego powodu, że cena nowego Epsona równa się cenom na drukarki laserowe. Tych ostatnich były na CeBIT setki. Współpraca „laserówek” z kserokopiarkami, scannerami, komputerem i naświetlarkami drukarskimi stwarza zupełnie nowe możliwości w poligrafii. Nic dziwnego zatem, że mniej i bardziej profesjonalny Desktop Publishing był jednym z najmocniej eksponowanych zastosowań mikrokomputerów.



Rodzinę PC reprezentowały zarówno potężnie wyglądające wieże, jak i portable.

Wśród drukarek dla amatorów brylowali ci sami co zwykle Japończycy. „Star” — najpopularniejsza dziś bez wątpienia firma w naszym kraju, a nr. 2 na świecie, zgodnie z zapowiedzią przedstawił kolorowa wersję popularnej LC-10. Ta drukarka, we wszystkich swoich odmianach (zaprezentujemy wkrótce test jej wersji na Commodore) ma szansę stać się naprawdę biurowym i amatorskim standardem, przynajmniej w naszym kraju. Przeszkodzić mu w tym będzie zapewne chciał Citizen, który również pokazał kilka interesujących propozycji drukarek średniej klasy. Citizen i Star były zresztą partnerami polskich handlowców z „Metronexu” i dyrekcji zakładów Mera z Błonia podczas rozmów na temat unowocześnienia rodzimych drukarek. Wieść niesie, że tym razem rozmowy leszcze nie przyniosły konkretnych rezultatów.

Polska na CeBIT reprezentowana była przez sporą grupę dziennikarzy, paru właścicieli firmy komputerowych szukających okazji do zakupu tanich PC oraz ewentualnej sprzedaży softwaru i przez wspomniany wyżej „Metronex”, który zainstalował nawet swoje stoisko. Niestety oferta na nim przedstawiona miała raczej wartość muzealną w zestawieniu z tym, co pokazali inni.

Ekipa „Bajtka” przyjmowana była gościnnie, chociaż, na ogół, pierwszy rzut oka rozmówców na jakość druku naszego pisma nasuwał im podejrzenie, że jesteśmy gazetką szkolną. Dopiero informacja o nakładzie wywoływała szok a uprzejmą obojętność zmieniała w zainteresowanie. A my, oczywiście, nie wyjaśnialiśmy nigdzie, że i tak „Bajtek” to jedno z najporządniej wydawanych w Polsce pism.

Do wrażeń z Hanowerskiej imprezy będziemy jeszcze zapewne nie raz wracać. Pokazała nam ona dobitnie czym jest nadchodząca era informatyczna i jak wiele nas, Polaków, od niej dzieli. To już jednak temat na oddzielne rozważania, jak mawiają politycy i naukowcy

Tekst i zdjęcia:
Grzegorz Onichimowski
Roman Poznański

```
var
  Prog     : file of byte;
  Txt       : text;
  Nazwa : string [14]:
  B : byte:
  i : integer;

procedure Stop;
begin
  writeln ('Koniec programu...'};
  halt
end;

begin
  ClrScr;
  writeln ('      -=-=- Dekompilator Turbo -=-=-');
  writeln ('Program ten pozwala na przeksztalcenie ');
  writeln ('programu w jezyku maszynowym do formy ');
  writeln ('tekstu procedury Inline akceptowanej ');
  writeln ('przez jezyk Turbo- Pascal.');
  writeln ('Program czyta plik (program wejsciowy) ');
  writeln ('XXXX.COM i tworzy odpowiadajacy mu plik');
  writeln ('wyjsciowy XXXX.INL.');
  writeln ('Aby dolaczyc otrzymany kod do programu w');
  writeln ('Turbo, nalezy w odpowiednim jego miejscu');
  writeln ('napisac:');
  writeln ('     <$I XXXX.INL> ');
  writeln ('Zatrzezenia:');
  writeln (' 1. Dekompilowany program powinien byc ');
  writeln ('relokowalny. ');
  writeln (' 2. W przypadku Turbo v.3.0 dekompilacja ma');
  writeln ('sens, gdy plik XXXX.INL nie zajmie wiecej ');
  writeln ('niz 64k.'};
  writeln ('W tych przypadkach program mozna uruchomic');
  writeln ('z wnetrza programu w Turbo przy pomocy');
  writeln ('procedury Execute.');
  writeln;
  writeln (Nazwa pliku dekompilowanego?');
  writeln ('<tylko ENTER = koniec programu> ');
  read (Nazwa}; writeln; writeln;
  if Nazwa = '' then Stop;
  i := Length(Nazywa};
  while (i   >  0) and  (Nazwa[i] <> '.') do
  i  := pred(i); if i <> 0 then Delete(Nazwa,  i,   14);
  {$I-}
  writeln ( Dekompilacja pliku '. Nazwa.   '.COM');
  Assign(Prog, Nazwa + '.COM');
  Assign(Txt, Nazwa + '.INL');
  writeln ('na plik tekstowy '. Nazwa.  '.INL');
  reset(Prog);
  if IOResuit <> 0 then begin
  writeln ('Nie moge znalezc pliku '.Nazwa.'.COM');
  Stop
end;
rewrite(Txt);
if IOResult <> 0 then begin
  writeln ('Nie moge utworzyc pliku '.Nazwa.'.INL');
  Stop
end;
write(Txt,  inline(');
i := 8;
whiie not eof (Prog) do begin
    read(Prog, B);
    write(Txt,  B);
    if B  >  99 then i  := i + 4
    else
    if B  >  9  then i  := i + 3
    else i := i + 2;
    if i > 70 then begin
      writeln(Txt);
      i := 0
    end;
    if not eof (Prog) then write (Txt, '/')
    else writeln(Txt.');')
  end;
  Close(Txt);
  Close(Prog};
  writeln ('O. K');
  Stop
end.
```

TURBO PASCAL
A JĘZYK MASZYNOWY

Niekiedy zachodzi potrzeba wykorzystania we własnym programie programu innego, zapisanego w innym języku programowania (np. asemblerze) albo wręcz dostępnego nam jedynie w postaci kodu maszynowego. Turbo Pascal częściowo udostępnia narzędzia pozwalające na włączanie „obcych" programów do programów w Turbo. Sa to:

1. Instrukcja external. Oto przykład jej użycia, procedurę XYZ;
external sfe00;
Słowo external informuje kompilator, że procedura XYZ zostanie skompilowana i umieszczona w pamięci niezależnie od kompilowanego programu w Pascalu. Liczba całkowita stanowiąca parametr instrukcji external określa adres początku tej procedury. Wykonanie procedury XYZ nie może zmieniać wartości rejestru SP (wskaźnika stosu).

2. Procedura Execute. Oto jeden z możliwych przykładów użycia (zakładamy, że zmienna F jest dowolnego typu plikowego):
Assign (F, 'A:TIMER.COM'); Execute (F);
Taka sekwencja Assign i Execute powoduje wykonanie znajdującego się na dysku A: programu TIMER. Ogólnie rzecz biorąc, wywołanie Execute (F) powoduje wykonanie programu zawartego na pliku dyskowym związanym ze zmienną F.
Procedura Execute jest bardzo przydatna, jeżeli chcemy w naszym programie użyć innego programu o dużej objętości (chociażby edytora). Co innego jednak, gdybyśmy chcieli w ten sposób dołączyć program niewielki, za to wywoływany wielokrotnie. Wówczas każdorazowe wczytywanie go z dysku nie ma większego sensu (gdy chcemy np. w ten sposób wyświetlić czas systemowy).

3. Czytelnik znający dobrze Turbo Pascal zauważy w tym miejscu, że jest proste rozwiązanie i takiego problemu, a mianowicie procedura (a może raczej instrukcja?) Inline. Pozwala ona na bezpośrednie włączenie kodu maszynowego do programu w Pascal-u. Oto przykład; nie ma on większego sensu od strony użytkowej, ale za to demonstruje wszystkie możliwości generowania kodu przez użytkownika:
iniine (123 / SFA / XyzDuze / *+10 / >01 /< 1234);
Z przykładu tego wynika wspomniana wyżej trudność terminologiczna: Inline wyposażono w składnię odpowiadającą z grubsza wywołaniu procedury. Procedurą to jednak nie jest, gdyż umieszczenie powyższego napisu w programie nie odpowiada wywołaniu żadnego podprogramu. Problemu tego nie będziemy tu rozstrzygali — ważniejsze od dyskusji terminologicznych jest wszakże sprawne posługiwanie się otrzymanym narzędziem.
Jak widzimy, rolę parametrów Inline spełnia ciąg liczb całkowitych, oddzielonych znakami W roli tych liczb mogą występować, poczynając od lewej:
a. Stałe całkowite w postaci dziesiętnej,
b. Stałe całkowite w postaci szesnastkowej,
c. Identyfikatory zmiennych. W miejscu wystąpienia takiego identyfikatora kompilator wstawia dwubajtowy adres zmiennej,
d. Stan licznika adresowego (PC) w momencie wykonania programu, reprezentowany przez znak gwiazdki. Jego użycie umożliwia adresowanie względne w obrębie kodu.
e. Wszystkie powyżej wymienione postacie liczb możemy łączyć w wyrażenie przy pomocy operacji + i -.
Dwa ostatnie wyrażenia w podanym przykładzie wiążą się z długością kodu generowanego dla każdego z nich. W normalnej sytuacji kompilator sprawdza, czy wartość liczby jest mniejsza od 256 i zależnie od wyniku sprawdzenia generuje jeden lub dwa bajty kodu. Nie dotyczy to adresów zmiennych, które zawsze są reprezentowane w postaci pary bajtów. Regułę tą możemy zmienić przy użyciu znaków mniejszości i większości. Pierwszy z nich oznacza, że należy wygenerować kod jednobajtowy, zawierający tylko młodszy bajt podanej liczby. Drugi informuje kompilator, że podana liczba ma być umieszczona w dwóch bajtach.
Tyle na temat instrukcji (procedury?) inline. Pomimo wielu opisanych powyżej udogodnień w jej stosowaniu, ma ona jedną podstawową wadę: trzeba znać kod maszynowy. Poza tym programowanie bezpośrednio w tym kodzie jest wyjątkowo trudne i podatne na błędy.
Idealnym rozwiązaniem byłoby połączenie Turbo Pascal-a z asemblerem. Zamieszczony obok program pokazuje, w jaki sposób rozwiązałem ten problem dla własnych potrzeb — jak dotychczas jestem z niego zadowolony.
Potrzebny fragment programu piszę w asemblerze, a następnie w odpowiedni dla tego języka sposób testuję i uruchamiam. Wynikiem tych operacji jest gotowa i sprawdzona wstawka do programu w Pascalu. Następnie asembluję ją do postaci kodu maszynowego, dbając uprzednio o to, aby kod ten był relokowalny.
Teraz pozostaje do zrobienia naprawdę niewiele. Otrzymany fragment kodu przetwarzam przy pomocy zamieszczonego obok programu do postaci wymaganej przez Turbo Pascal, tzn. procedury (instrukcji?) Inline z odpowiednim ciągiem liczb.
Otrzymany w ten sposób tekst (zawarty na nowym pliku) dołączam do tekstu programu bądź bezpośrednio, bądź też przy pomocy dyrektywy kompilacji SI.
W podobny sposób można także przekształcać gotowe programy (tzn. takie, gdzie dysponujemy tylko ich postacią skompilowaną) w część programu w Pascal-u. Obowiązuje wówczas w dalszym ciągu wymóg relokowalności kodu. Dodatkowo Turbo Pascal nakłada na tekst źródłowy programu wymóg nieprzekraczania objętości 64 KB. Biorąc pod uwagę, że tekst zawarty w Inline jest średnio co najmniej 4 razy dłuższy od generowanego kodu, użycie tej metody jest sensowne co najwyżej dla programów liczących 10-12 KB. Nie jest to dużo, ale dla programów większych wciąż mamy dostępną procedurę Execute.
Opisaną metodę polecam użytkownikom Turbo Pascal-a do ewentualnego spożytkowania. A może ktoś wymyśli wygodniejsze rozwiązanie?

Marek Wyrwidąb



ZUPA Z GWOŹDZIA

(O SMAKU MNIEMANOLGII STOSOWANEJ)

Dziś w dalszym ciągu zajmiemy sie programami i tym, co jest z nimi nierozerwalnie związane — błędami. To drugie może zainteresować nie tylko programistów.

Programując w zeszłym miesiącu algorytm Homera stanęliśmy przed problemem zapisania danych w pamięci. Dla różnych danych potrzebne są tablice o różnych długościach. Najlepiej byłoby deklarować tablice w czasie działania programu — po wczytaniu wielkości danych, ale większość języków programowania nie dopuszcza takiego rozwiązania[*]. Tablica może być zdefiniowana tylko raz, często jej zakres musi być ustalony już w momencie tłumaczenia programu, czyli musi być stałą (np. w Pascalu czy w FORTRANIE).

Typowym rozwiązaniem tego problemu jest zadeklarowanie tablicy mogącej pomieścić każde z oczekiwanych danych, a następnie dla każdego zadania wykorzystujemy początkowy fragment tej tablicy. Popatrzmy jeszcze raz na fragment programu liczącego wartość wielomianu:

```
A:array[0...100] of real;
...
readln (n)
...
for i:=0 to n do A[i]:=i; w:=A[n];
for i:=n—1 down to 0 do w:=w * x+A-[i];
```

Oczywiście, podobną metodę można zastosować do tablic wielowymiarowych.

Skoro napisaliśmy program, to trzeba go przetestować. Zapewne zrobimy to tak: weźmiemy kilka wielomianów niewielkiego stopnia, obliczymy na papierze czy kalkulatorze ich wartość i sprawdzimy, czy program daje taki sam wynik. Jeśli tak, uznamy że mamy dobry program do obliczania wartości wielomianów.

Autor takiego programu może się cieszyć, ja też. Bo dostałem właśnie świetny przykład na program, który jest błędny ale może długo działać bezbłędnie zanim pewnego pięknego dnia nie padnie. Szczególnie długo będzie działał poprawnie, gdy testować i używać będzie go sam autor. Ma on bowiem zakodowane w podświadomości, że NIE WOLNO podawać w danych wartości n większej niż 100[**]. Oczywiste jest, że gdy n = 100 to program będzie odwoływał się do nieistniejących elementów tablicy i trudno oczekiwać dobrych rezultatów.

Możliwe są tu dwa warianty rozwoju sytuacji, w zależności od języka programowania i translatora. Program może być liczony z kontrolą indeksów, oznacza to, że przy każdym wystąpieniu odwołania A[i] sprawdzane jest w momencie wykonania, czy wartość i nie przekracza dopuszczalnej (określonej w definicji tablicy) wartości. Jeśli przekracza, sygnalizowany jest błąd wykonania. Jest to wariant optymalny — błąd nie może przejść niezauważony — ale też i kosztowny. Wykonanie programów często korzystających z tablic zostaje znacznie spowolnione, gdyż każde odwołanie wymaga kontroli. Niektóre translatory w ogóle nie przewidują kontroli indeksów. Niektóre pozwalają kontrolę tę włączać lub wyłączać przez wybór odpowiednich opcji w momencie translacji. Często postępujemy tak: w czasie testowania liczymy program z kontrolą indeksów, do eksploatacji przekazujemy wersję kodu bez tej kontroli.

Pójdźmy teraz drugą, znacznie ciekawszą, ścieżką[***]: kontrola indeksów wyłączona, więc program „bez skrupułów” będzie sięgał do nieistniejących elementów tablicy. Zaczynym od eksperymentu, deklarujemy tablicę nie [0...100] tylko np. [0...2] (wtedy dopuszczalne n=2), żeby łatwo móc ręcznie sprawdzać wyniki i puszczamy program. Wynik na pewno mianu wstawimy do B jakieś wartości, to nasz program zaczyna chodzić w straszne buraki — nie sygnalizuje błędu a liczy źle! Mało tego, nie wszystkie przypadki, tylko niektóre! Mało tego, liczy źle przypadki, które przedtem liczył dobrze! Chociaż jedyną zmianą w programie było wstawienie instrukcji wczytującej wartości do tablicy B, a więc instrukcji nie mających nic wspólnego z wielomianem A! Dlaczego zaczął liczyć źle?

Zaczynamy podejrzewać wszystko: komputer, kompilator, ba, ogłaszamy, że jesteśmy w posiadaniu niezbitego dowodu, że translator Pascala nie działa poprawnie. Ale robimy to pochopnie, zbyt pochopnie.

Wróćmy do zarzutów i przyjrzyjmy się im uważnie: kompilator (lub komputer) jest zły, bo dodaliśmy instrukcje, w których nawet nie występuje żaden z elementów wielomianu, a obliczenie wielomianu zostało popsute Oczywistą nieprawdą (może nie na pierwszy rzut oka, ale oczywistą) jest stwierdzenie, że dodane instrukcje nie używają elementów wielomianu. Dlaczego nieprawdą? Dlatego, że używają! Używają, tych elementów, które bezprawnie zostały umieszczone w obszarze pamięci przeznaczonym na tablicę B. Bezprawnie, ale z naszej, a nie komputera winy, przecież podaliśmy niedopuszczalnie dużą (dla tego programu) wartość n. Popatrzmy na rysunek, obrazujący rozmieszczenie elementów tablicy w pamięci operacyjnej:

A[0] A[1] .... A[99] A[100] B[0] B[1] ... B[100]

Jeżeli n jest większe od 100, pętla operująca na wielomianie odwołuje się do nieistniejących elementów A[101], A[102], itd., a więc bezczelnie włazi na elementy tablicy B. Dopóki tablicy B nie używamy przed obliczeniami z wielomianem A, uchodzi nam to na sucho. Ot i cała tajemnica jak zwykle okazuje się, że błędu trzeba było szukać we własnym programie. Podkreślmy jednak: cały proces obliczania wartości jest poprawny, błędem jest tylko dopuszczenie do obliczeń niepoprawnych, za dużych danych. Więc szukanie błędu w samym liczeniu, czy też w schemacie Homera będzie bezskuteczne.

Mało tego, wyobraźmy sobie sytuację jeszcze gorszą: dane opisujące wielomian nie są podawane przez operatora, lecz powstały jako wynik pracy innego programu i są przekazywane z programu do programu w ramach rozbudowanego systemu informatycznego, rozwiązującego jakiś duży problem. W programie produkującym wielomiany, które mają być obliczane, jest niewielki błąd, który powoduje, że raz na pewien czas powstaje wielomian o stopniu większym od największego dopuszczalnego. Wskazuję taki przypadek jako szczególnie złośliwy, gdyż błąd nie dość, że trudny do zauważenia (pojawia się nie za każdym razem), jest trudny do zlokalizowania — ostateczny błędny wynik otrzymujemy z poprawnego podprogramu, nie z tego, w którym błąd rzeczywiście powstał.

Oprócz szczegółów technicznych (zakresy tablic, kontrola danych itd.) postarajmy się wyciągnąć z opisanej sytuacji także bardziej ogólną naukę: błąd w programie może być naprawdę bardzo trudny do znalezienia, może nawet występować daleko od miejsca, w którym wykryliśmy objawy. Zawsze jednak należy go szukać właśnie tam, w programie, nigdy zaś w błędnym działaniu komputera. Np. w opisanym wyżej przypadku doświadczony programista poradzi sobie dobrze i szybko, wstawiając na początku programu kontrolny wydruk tego, co dostaje jako dane. A jeszcze bardziej doświadczony wcale nie będzie czekał, aż pojawi się błąd, tylko zrobi to, co trzeba było zrobić na początku — wstawi kontrolę poprawności danych już w chwili pisania programu.

Napiszmy więc wreszcie obiecany już miesiąc temu program: program hor (input, output):

```
const
    nmax=20; var
x,w: real;
    n,i :integer;
    A: array[0..nmax] of real;
begin
writelnfPodaj stopień n);
readln(n);
if n>nmax then
    writeln('Błąd — max. dopuszczalna
wartość n wynosi', nmax) else
    begin
    {generacja danych testowych}
for i:=0 to n do A[i]:=i;
writeln('Podaj x'); r
eadln(x);
(początek obliczeń)
w:=A(n);
for i: = n — 1 downto 0 do
w: = w*x+A[i];
writelnfDla x=',x', wartość wielomianu
wynosi: w)
    end
end.
```

Zupa z gwoździa najlepiej smakuje, gdy dodać do niej trochę innych przypraw. Na podobnej zasadzie powyższy nieduży program jest tematem całego ~~artykułu. Bądźmy jednak oszczędni.~~ Za miesiąc spróbujemy dodać innych przypraw i ugotować z tego samego gwoździa zupę o innym smaku.

Andrzej Pilaszek

***) Możliwość tzw. dynamicznego deklarowania tablic istnieje np. w języku programowania ALGOL 60 i jego mutacjach. Język ten, dziś już częściowo zapomniany, był przez wiele lat niezwykle popularny.**

****) Jest to jedno z uzasadnień propozycji, aby prosić o pomoc przy testowaniu kogoś nie znającego programu.**

*****) Jest to bardzo interesujące, dopóki prowadzimy rozważania teoretyczne. Ludzie, którym sytuacja podobna do opisanej dalej przydaży się w praktyce wcale nie są zachwyceni. Na ogół to co mają do powiedzenia zupełnie nie nadaje się do druku.**



RANDKA Z FORTH-EM

Łatwość korzystania z Basicu, który zgłasza się zwykle natychmiast po włączeniu zasilania mikrokomputera, nie jest okolicznością zachęcającą do poznawania innych języków programowania. Jednakże już po kilkunastu dniach zabawy z Basicem dostrzeżemy jego wady i dojdziemy do wniosku, że brakuje w nim środków zapewniających sprawną realizację niektórych zadań. Należy wtedy pomyśleć, czy nie warto czasami przestudiować podręcznika jakiegoś innego języka.

Zaznajomienie się z ciekawym i bardzo efektywnym narzędziem programowania umożliwi nam książka Jana Ruszczyca "Poznajemy FORTH", wydana przez SOETO. FORTH należy do języków strukturalnych i jest bardzo popularny na świecie. Jego przydatność potwierdziły liczne programy wymagające szybkiego przetwarzania danych. Użyto go między innymi w radioastronomii, do sterowania robotami przemysłowymi. Pomógł również w uzyskaniu dobrze znanych wszystkim kinomanom efektów w filmach science fiction z serii "Gwiezdne wojny" oraz opracowaniu wielu gier arkadowych dla komputerów Atari.

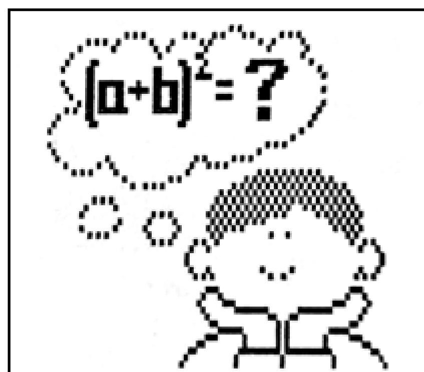
Książka przeznaczona jest nie tylko dla profesjonalistów. Wyodrębniono w niej część zawierającą wstępne informacje o języku, jego historii i zastosowaniach. Zasady programowania wykładane są krok po kroku, dzięki czemu ich opanowanie możliwe jest nawet dla laika. Autor umiejętnie zaprezentował odmienność rozwiązań użytych w FORTH-cie, które niewątpliwie zaszokują programistów przyzwyczajonych do Basica. Główne cechy charakterystyczne to: oparcie języka na podstawowej jednostce programowej zwanej słowem, możliwość grupowania słów w słowniku, posługiwanie się w obliczeniach stosem oraz odwrotną notacją polską. Cechy te sprawiają, że program napisany w FORTH jest zbliżony szybkością działania do programów utworzonych przy pomocy asemblera.

Druga część koncentruje się na praktycznym zastosowaniu poznanych struktur języka, kompilacji, operacjach wykonywanych na tablicach i działaniu operatorów logicznych. Zawarte tu informacje o współpracy ze stacją dysków, asemblerze stanowiącym integralną część systemu i edytorze są niezbędną podstawą do rozpoczęcia samodzielnego programowania. Przy okazji dowiadujemy się, że edytor FORTH może być wykorzystany do pisania tekstów lub programów w języku C. Książkę zamykają przykłady prostych gier oraz aneksy i bibliografię, opisem edytora, komunikatami błędów i spisem słów występujących w FORTH.

(j.j)

Jan Ruszczyć, "Poznajemy FORTH", SOETO, Warszawa, ul. Hoża 50, 1987, wyd. I, nakład 3000 egz., cena 950 zł.

PIERWSZY MATEMATYCZNY KONKURS BAJTKA



Zadanie: Napisać program, który w ciągu 60 sekund obliczy i przechowa w tablicy więcej liczb pierwszych.

Def.: Liczba pierwsza to taka liczba naturalna, która dzieli się bez reszty, TYLKO sama przez siebie i przez liczbę 1. Do liczb pierwszych zaliczamy: 1, 2, 3, 5, 7, 11, 13...

Odpowiedzią ma być przysłany program pisany wyłącznie w języku BASIC. Należy również do programu dołączyć dokładny opis działania programu. Opis rozwiązania problemu i schemat blokowy oraz podać typ komputera, na którym zadanie zostało rozwiązane. Należy podać otrzymaną ilość liczb pierwszych.

UWAGA!!! Wszystkie programy (pretendujące do wygranej) będą wpisane na jeden komputer po to, aby wygrał algorytm, a nie szybszy procesor. Każdy więc ma równe szanse. Liczą się pomysły.

Regulamin:

1) Program ma być napisany przy użyciu języka BASIC. Dopuszczone są następujące instrukcje i funkcje: PRINT GOTO, GOSUB, RETURN, IF.. THEN, LET, DIM, FOR.. NEXT...STEP, AND, OR, NOT RND, ABS, INT, SQR, EXP ATN, COS, SIN, TAN, LOG (logarytm naturalny), PI (liczba Pi), SGN, +, -, *, /, UWAGA. Przy podstawianiu obowiązkowe jest używanie rozbudowanej instrukcji LET.
2) Obliczone liczby pierwsze mają być przechowane w jawnie określonym wektorze (tablicy) i nie muszą być w trakcie obliczeń wyprowadzane na monitor.
3) Program zaczyna się od linii o numerze 1 z przyrostem linii co 1.
4) Program zawiera po jednej instrukcji w linii.
5) Czas 60 sekund mierzony jest od momentu wykonania pierwszego obliczenia (nie jest brany pod uwagę czas potrzebny na zadeklarowanie tablicy, dlatego prosimy o dokonanie deklaracji na początku programu i zaznaczenie początku obliczeń np. przez PRINT" START". UWAGA. Jedyną instrukcją dopuszczoną przed PRINT„START" jest instrukcja DIM.

Redakcja ze swej strony, by zachęcić do rywalizacji, zobowiązuje się podawać najlepsze dotychczasowe wyniki w kilku kolejnych numerach BAJTKA przed definitywnym rozstrzygnięciem konkursu.

CZYTELNIKU "BAJTKA"

UWAGA! UWAGA!

Jeśli chciałbyś stać się posiadaczem kompletnego rocznika '87 naszego pisma lub brakuje Ci pojedynczych egzemplarzy z ubiegłego roku napisz do nas!

Przesyłamy zaległe egzemplarze wszystkim tym, którzy przyślę nam przekaz (rocznik— 1200 zł + 100 zł za przesyłkę) lub zamówią przesyłkę za zaliczeniem pocztowym. Prosimy tylko — nie wysyłajcie pieniędzy w kopertach

UWAGA!

WPHW O/DĄBROWA GÓRNICZA SKLEP NR 163

„ELEKTRON”

41-300 Dąbrowa Górnicza
ul. Sobieskiego 17

Jeden z pierwszych i najtańszych uspołecznionych punktów sprzedaży sprzętu minikomputerowego prowadzi skup i sprzedaż:

- minikomputerów 8 bitowych

		(ceny sprzedaży)
np. Spectrum 128K+2	—	400.000,-
Atari 65XE + mag.	—	350.000,-
130XE + mag.	—	420.000,-
520ST + SM124	—	1.800.000,-
Amstrad CPC464 z ziel.m.	—	520.000,-
CPC6128zziel.m.	—	950.000,-
Commodore C128	—	500.000,-

- drukarki (z kablami)

np. DMP 2000	—	650.000,-
STAR NL10	—	830.000,-
SC 15	—	1.150.000,-
NX15	—	1.250.000,-

- systemy minikomputerowe PC XT/AT w dowolnej konfiguracji
- urządzenia peryferyjne
- sprzęt video i CTV

np. magnetowid SANYO 3100EE	—	980.000,-
JVC HR-D210EE	—	1.150.000,-
telewizor SANYO 21'	—	1.150.000,-
SANYO 26'	—	1.650.000,-
Panasonic 26'	—	1.750.000,-

- systemy sprzęt i wyposażenie: dyskietki, kable, kasety, pojemniki, elementy elektroniczne, urządzenia dodatkowe

Sklep prowadzi sprzedaż pozarynkową.

Udziela gwarancji i zapewnia serwis poprzez PPT "ABM" Sp. z o.o. w Dąbrowie Górniczej, ul. Czerwonych Sztandarów 94.
Zapraszamy do sklepu w godz. 10.00—18.00 Dąbrowa Górnicza, ul. Sobieskiego 17, tel. 62-23-71.

WOJEWÓDZKIE PRZEDSIĘBIORSTWO HANDLU WEWNĘTRZNEGO ODDZIAŁ W TYCHACH

VIDEOBIT

43-100 Tychy, aleja ZMP 77
tel. 27-69-75

— minikomputery 8-bitowe (Atari, Commodore, Schneider-Amstrad)
— minikomputery 16-bitowe kompatybilne z IBM PC
— drukarki 10" i 15" firm STAR, EPSON, AMSTRAD
— magnetowidy
— kamery video
— anteny satelitarne
— aparaturę badawczo-naukową

Zapewniamy o atrakcyjnych cenach.

MICROMAN

Programy na Atari 800 XL, Spectrum 48 KB, Commodore 16/116/+4 na miejscu lub za zaliczeniem pocztowym. Informacje za załączeniem koperty i znaczka pocztowego. 40-181 Katowice, ul. Osikowa 66, tel. 58-51-06.

ATASERW

39-460 Nowa Dęba, skr. poczt.
Tel. TARNOBRZEG 46 22 58
oferuje świetne rozwiązania sprzętowe do ATARI:
1. TOP DRIVE 1050 (lepszy niż HAPPY)
2. INTERFEJS CENTRONICS
3. ROZSZERZENIA PAMIĘCI DO 128 i 256KB
4. PIÓRO ŚWIETLNE
5 BASIC XE KARTRIDŻ
6. PROGRAMATOR EPROM
7. KARTRIDŻE Z DOWOLNYM PROGRAMEM

Informacje po otrzymaniu koperty zwrot--nej

NAJSZYBCIEJ! NAJPEWNIEJ! NAJTANIEJ!

STUDIO KOMPUTEROWE "HOOZY", Sopot, Boh. Monte Cassino 21D, kod 81-706 udostępnia w największym wyborze oprogramowanie i literaturę do

ATARI

— unikalne wersje kasetowe programów dyskowych!
— naszą dewizę — jakość!

Wstąp lub napisz — informacje pocztą gratis po otrzymaniu koperty zwrotnej oraz znaczka.

ATARI BASIC, wersja polska. Cena 2000 zł. Zamówienia przekazem.
Żurko, 43-100 Tychy, Nałkowskiej 24/10

● ● ● ● ● ● ● ● ● ● ● ● ●

● ATARI ●

Szeroki wybór oprogramowania na kasetach i dyskietkach.
- **co piąty program bezpłatnie**
- **gwarancja jakości**
- **rachunki**
- **katalogi gratis**

ATR-SOFTWARE
66-542 Zwierzyn P-1

UDOSKONALENIA
PROGRAMOWE I SPRZĘTOWE
DLA WSZYSTKICH MODELI ATARI
ORAZ KOMPUTERÓW PRACUJĄCYCH POD SYSTEMEM MS-DOS
WYSYŁA POCZTĄ
agencja mikro-komputerowa *amico*
41-200 Sosnowiec
K-91

Dla programujących w języku Atari Basic lub Turbo Basic XL program:

TEKST KODER

pozwalający na pełne wykorzystanie możliwości graficznych komputera we własnych programach. Opis i programy Demo gratis, po przesłaniu kasety i znaczków. R. Pantoła, 37-450 Stalowa Wola, skr. poczt. 1594.

ZX SPECTRUM

Naprawy komputerów
Interfejsy do joysticków

ATARI

Interfejs magnetofonu
Możliwość wysyłki pocztą.
Jerzy Dymecki, ul. Meissnera 14 m 1, 03-982 WARSZAWA
tel. 15-93-38 po południu.

STUDIO KIJOWIANKA
AMSTRAD ● ATARI XL, XE, ST ● COMMODORE 64, 128
Poleca literaturę i programy na kasetach i dyskach. Warszawa, ul. Targowa 26. Rachunki oraz wysyłka pocztą. Informacje za załączeniem koperty i znaczka.

TELERADIONAPRAWA
W-wa 610-38-77
poleca wejścia monitorowe
dekodery PAL SECAM NTSC

GRY, PROGRAMY UŻYTKOWE, OPISY NA ATARI XL/XE OFERUJE "MIKROFAN", 45-064 OPOLE 1, SKR. POCZT. 158 (informacje za załączeniem znaczka).

Sprzedam ATARI (520 STM) 354, 44-100 Gliwice, Michałowskiego 20, tel. 31-55-84

Wymiana programów —
— ZX SPECTRUM
Fryderyk Badura
44-200 Rybnik
Dworcowa 5/41.

	GIEŁDA BAJTKA (tys. zł)	PEWEX BALTONA (USD)	RFN (śred.) (DM)
SINCLAIR			
ZX81	35	–	39
ZX Spectrum 48 kB	120	115	110-130
ZX Spectrum Plus	140	–	180-210
ZX Spectrum 128 + 2	240	–	220
Drukarka SEIKOSHA GP50S	90	–	99
TIMEX 2048	140	146	–
Joystick	4,5-7	–	4-6
COMMODORE			
C–64	210	219	299
C–128	300	299	450
C–128D	750	–	900
Amiga 500	1.4 mln	–	1000
Magnetofon 1531	45	48	30
Stacja dyskietek 1541	210	–	399
Stacja dyskietek 1571	240	299	460
Drukarka GP-500	190	–	149
Dyskietki 5 1/4 (średnia jakość)	0.8–1.5	3.5	0.3-1,5
ATARI			
65 XE	175	125	65
130 XE	260	199	105
Stacja dyskietek 1050	230	187	350
Atari 520 STM St. Dysk. 0.5Mb	1.1 mln	798	800
AMSTRAD			
464 z mon. monochromat.	280	–	350
6128 z mon. monochromat.	450	–	700
6128 z monitorem kolorowym	550	–	1000
Dyskietki 3"	5	–	6-9
Stacja dyskietek 3" do 464	380	–	399
PC 1512 SD MD	1.0 mln	–	999

Giełdy Bajtka odbywają się przy ul. Grzybowskiej 35 (Szkoła Podstawowa nr 25) w każdą sobotę i niedzielę od 14.00 do 19.00

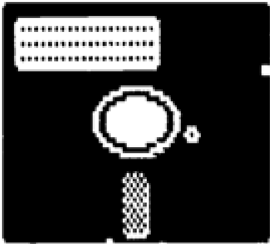
WYZWANIE HACKEROM

Piractwo komputerowe, które jako zjawisko należałoby tępić, dla stałych bywalców giełd jest jedynym źródłem zdobywania oprogramowania. Bo też kogo byłoby stać z kieszonkowego od rodziców na kupno w renomowanych firmach dowolnego programu, chronionego prawami autorskimi? Natomiast operatywni „włamywacze komputerowi” nie ponosząc kosztów autorskich honorariów mogą dostarczyć oprogramowanie po całkiem dostępnych cenach.

Tymczasem japońska firma Nippon Telegraph and Telephone (NTT) zapowiedziała wprowadzenie wiosną tego roku układu scalonego, generującego i sprawdzającego szyfr, dzięki któremu można używać systemów komputerowych, zabezpieczonych w ten sposób przed dostępem do informacji osób niepowołanych.

Czas pokaże, na ile zabezpieczenie okaże się skuteczne i czym odpowiedzą giełdowi „fachowcy”.

(gr)



INDYWIDUALNY BANK DANYCH

Alicja Wilczek, lat 16. Posiada mikrokomputer Timex 2048 kompatybilny z ZX-Spectrum. Oprogramowanie: gry i programy użytkowe m.in. LOGO. Zainteresowania: gry komputerowe, muzyka rozrywkowa i informatyka. Adres: 41-922 Bytom, ul. Kalei 32/8.

Marcin Łapkowski, lat 14. Posiada ATARI 1040 STF oraz ATARI 520 STF. Oprogramowanie: około 50 gier oraz wiele programów użytkowych, posiada również wiele pism francuskich o Atari. Adres: 22 rue de la BEAUME, 38-180 SEYSSINS, Grenoble, France; oraz w Polsce 41-804 Zabrze, ul. Zaolziańska 3/4.

Paweł Świątek, lat 11. Posiada ATARI 130 XE, stację dysków ATARI 1050 oraz drukarkę LEGEND 808 i monitor kolorowy. Oprogramowanie: ponad 100 gier i około 25 programów użytkowych, w tym dużo graficznych do drukarki. Chętnie nawiąże kontakt z użytkownikami ATARI, którzy posiadają stację dysków. Adres: 31-106 Kraków, ul. Wygoda 11/2.

Radosław Laskowski, lat 14. Posiada mikrokomputer C-64, monitor, magnetofon, joystick, pióro świetlne, organki. Oprogramowanie: ponad 50 programów. Proponuję wymianę doświadczeń oraz programów. Adres: 83-400 Kościerzyna, ul. T. Rogali 30.

Ireneusz Kapler, lat 12. Posiada ZX-81 + 16 kb. Oprogramowanie: gry, programy użytkowe i edukacyjne. Proponuje wymianę programów. Adres: 43-126 Katowice, ul. Bielska 10/4.

Andrzej Hofman, lat 27. Posiada ZX Spectrum 48 kb. Oprogramowanie: programy użytkowe, edukacyjne, kilkaset gier. Proponuje wymianę doświadczeń. Adres: 43-300 Bielsko-Biała, ul. I Armii Wojska Polskiego 4/41.

Jarosław Radzikowski, lat 32. Posiada mikrokomputer ATARI 800 XL z magnetofonem XC-12. Oprogramowanie: gry i programy użytkowe. Proponuję wymianę oprogramowania. Adres: 71-540 Szczecin, ul. Cedyńska 26/5.

Rafał Świerczyński, lat 11. Posiada mikrokomputer Amstrad CPC 464, monitor zielony, joystick. Oprogramowanie: gry oraz programy użytkowe. Proponuje wymianę oprogramowania oraz doświadczeń. Adres: 02-777 Warszawa, ul. Rogozińskiego 15 m 54.

Piotr Strzyżewski, lat 17. Posiada ATARI 800 XL, stację dysków turbo 1050, magnetofon XC-12. Proponuje wymianę oprogramowania. Adres: 89-203 Rynaczewo, ul. Szubińska 29, woj. bydgoskie.

Karol Rupik, lat 14. Posiada mikrokomputer Commodore 64 wraz z całym wyposażeniem oraz około 300 programów głównie gier. Proponuje wymianę oprogramowania. Adres: 41-908 Bytom, ul. Felińskiego 173/1.

Przemysław Palmowski, lat 17. Posiada mikrokomputer Amstrad/Schneider CPC-464. Pragnie nawiązać kontakt w celu wymiany gier i programów użytkowych. Adres: 72-510 Międzyzdroje, ul. Kolejowa 28/5.

Drogi Bajtku!

Na listy czytelników odpowiada Marcin Waligórski

Zająłem się niedawno problemem, a raczej ciekawostką matematyczną, tzw. magicznym kwadratem, w którym sumy liczb zawartych w wierszach są sobie równe i równe sumom liczb w kolumnach i na obydwu przekątnych.

Udało mi się napisać program, który bezbłędnie liczy (znajduje — przyp. M. W.) wyżej wymienione kwadraty dla boku od 3 do 87 (dla 89 rezultatem działania programu jest komunikat „Out of memory”).

Niestety, program wykonuje się tylko dla kwadratów o nieparzystym boku. Mam w związku z tym prośbę. Proszę o informację, czy w ogóle istnieją kwadraty o bokach parzystej długości? Wtedy jako dłużnik uzupełnię swój program i przyślę w finalnej postaci. Jeżeli nie, chętnie dostarczę go w obecnej postaci, o ile redakcję to zainteresuje. Program napisałem w Basic-u, więc liczy się dość wolno.

Bogdan Kurowski
Potok 6/3 68-132 Przewóz

Dla programisty-nowicjusza rozwiązanie postawionego powyżej problemu może wydawać się banalne. Wystarczyłoby przecież utworzyć kwadratową tablicę o żądanym boku, a następnie, próba za próbą, generować wszystkie możliwe kombinacje liczb ją wypełniających i sprawdzać, czy otrzymaliśmy kwadrat magiczny poprzez porównanie odpowiednich sum. Algorytm ten miałby jeszcze tę dodatkową zaletę, że moglibyśmy w ten sposób znaleźć nie tylko jeden, ale wszystkie kwadraty o danym boku i wartościach liczb zawartych w pewnym skończonym przedziale.

Pomysł prosty, tyle że... niewykonalny. Złożoność czasowa takiego algorytmu wyraża się wzorem (np)2, gdzie p jest długością boku kwadratu, zaś n — długością rozpatrywanego przedziału. Ten wzór sprawia, że już dla niewielkich zadań tego typu uruchamianie programu nie ma najmniejszego sensu; przy najszybszych obecnie produkowanych komputerach i tak na wynik trzeba byłoby czekać latami...

Ten krótki wstęp dedykuję czytelnikom, którzy się nad tym problemem nie zastanawiali, nie zaś p. Kurowskiemu, który, jak wnoszę z danych podanych w liście, już zdążył podane rozwiązanie świadomie odrzucić.

A teraz odpowiedź na postawione pytanie: tak, kwadraty magiczne o parzystej długości boku istnieją. Nawet rzekomo najstarszy znany europejski kwadrat magiczny, uwieczniony na jednym z obrazów Durera, datowanym 1514, ma bok długości 4:

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Przykład ten zaczerpnąłem z doskonale zapewne znanej młodszym czytelnikom „Bajtka” książki Sz. Jeleńskiego „Lilavati”, wyd. VIII, WSziP 1982. Autor poświęca w niej cały III rozdział figurom magicznym — a w szczególności algorytmów tworzenia kwadratów magicznych. Algorytmów tych jest wiele, lecz rzecz ciekawa, że żaden z nich nie działa jednocześnie dla dowolnej liczby parzystej, jak i nieparzystej — i tu leży zapewne źródło Pańskich kłopotów. Zachęcam do lektury i do podzielenia się wynikami swojej pracy z redakcją i czytelnikami „Bajtka”.

Od kilku miesięcy uczę się programować w Turbo Pascal-u. Ostatnio natrafiłem na problem, z którym nie mogę sobie poradzić. Chciałbym napisać jakąś grę zręcznościową i potrzebuję użyć funkcji, której wartością byłby ostatnio wciśnięty klawisz klawiatury. Wiem, że w Basic-u funkcją taką jest INKEYS, a w Hisoft Pascal-u INCH. Czy jest funkcja o podobnym działaniu w Turbo Pascal-u na IBM PC? Jeżeli jest, to jaka? Jeżeli jej nie ma, to jak ją zastąpić innymi instrukcjami (lub nawet kodem maszynowym)? W Turbo jest jedynie instrukcja KeyPressed, ale ona tylko sprawdza, czy został wciśnięty jakikolwiek klawisz, a to mi nie wystarcza. Proszę o pomoc.

Paweł Dec
ul. Chełmońskiego 8/1 35-111 Rzeszów

Turbo Pascal nie zawiera analogicznej funkcji standardowej, ale jej odpowiednik jest łatwo napisać samemu. Jest to o tyle proste, że Turbo Pascal jest wyposażony w standardową zmienną plikową **Kbd**, która odzwierciedla stan klawiatury, z pominięciem układu wejścia — wyjścia Pascal-a. Pliku **Kbd** można używać tylko do czytania. Operacja **read (Kbd, Znak)** odpowiada nadaniu zmiennej Znak typu char wartości odpowiadającej aktualnie wciśniętemu klawiszowi. Oto, jak rozwiązanie to można wykorzystać dla utworzenia odpowiednika znanej z dialektu Hisoft funkcji Inch:

```
function Inch: char; var Z: char; begin
read (Kbd, Z); Inch : = Z end;
```

Ponieważ wartość funkcji Inch nie jest określona w momencie, w którym nie jest wciśnięty żaden klawisz, powinniśmy jej używać tylko w połączeniu z funkcją KeyPressed:

```
if KeyPressed then Klawisz : = Inch;
```

lub, jeżeli chcemy, aby komputer oczekiwał na wciśnięcie pewnego klawisza:

```
repeat until Keypressed;
Klawisz : = Inch;
```

Jestem posiadaczem komputera Atari. W „Bajtku” nr 7/19 z lipca 1987 roku zamieszczony był program „Zmiana napisów w programach”. Po jego wpisaniu wystąpił błąd 9 w linii 40. Komputer posiadam od niedawna i jeszcze niewiele umiem, więc nie udało mi się go poprawić. Mam jeszcze jedną prośbę; komputer Atari ma spore możliwości muzyczne, ale pomyślcie sami, dać 500 zł (!!!) za byle program? Proszę, wydrukujcie choćby krótki program, który dałby dostęp do szerokich możliwości Atari.

Krzysiek
(nazwisko i adres do wiadomości redakcji)

Odpowiem w imieniu całej redakcji: „Bajtek” nigdy nie uważał i nie uważa, że programy komputerowe powinno się otrzymywać za darmo; dlatego też sprzeciwiamy się (niezbyt zresztą jasnemu) stanowisku czytelnika. Kilka lat niepodzielnego panowania piratów na naszym rynku i ciągły, coraz bardziej dotkliwy brak odpowiednich norm prawnych zdążył już stworzyć masę trudności producentom polskiego oprogramowania i duże spustoszenia w psychice polskich nabywców tegoż.

Do czego dążymy? Do normalnej sytuacji, w której przeciętny użytkownik komputera nie kupuje „byle programów”, a tylko te wybrane, najlepsze, za które płaci po 500 zł (a być może więcej, jeżeli oprogramowanie jest tego warte). Z drugiej strony chcemy, by polski producent oprogramowania mógł bez obawy sprzedawać swój produkt masowo i bez sztucznego śrubowania w górę ceny w przeświadczeniu, że w krótkim czasie wpływy ustaną na skutek kradzieży jego pomysłów i pracy. Proszę mi wierzyć, drogi czytelniku, że taka sytuacja jest jedyną szansą na rozwój polskiej produkcji oprogramowania dla potrzeb kraju; szansą, którą od kilku lat zaprzepaszczamy.

Proszę o odpowiedź na 3 pytania dotyczące komputera ZX Spectrum.

1. Na czym polega arytmetyka BCD?

2. Czy powrót z przerwania maskowalnego można wykonać przez RET, a nie przez RETI? Czym różnią się te rozkazy?

3. Jak zaprogramowany jest podkład dźwiękowy w grach (np. wybuch, strzał)?

(nazwisko i adres do wiadomości redakcji)

lSkrót BCD nie dotyczy arytmetyki, ale sposobu kodowania liczb całkowitych (Binary Coded Decimal — dwójkowo kodowany zapis dziesiętny). Kodowanie liczby polega na tym, że jej zapis dziesiętny przekładamy jedynie częściowo na układ binarny. Mianowicie kodujemy dwójkowo każdą cyfrę zapisu dziesiętnego osobno. Jak łatwo obliczyć, dla zakodowania jednej cyfry wystarczą nam 4 bity, zatem w jednym bajcie mieścimy dwie cyfry dziesiętne. Wadą tego systemu jest pamięciochłonność. W jednym bajcie możemy zmieścić liczbę całkowitą z zakresu 0..99, wobec 0..255 przy zwykłej konwersji binarnej. Zaletą jest zgodność zapisu liczby z naszą intuicją, i co za tym idzie — nieco prostsze staje się zadanie zapisu procedur arytmetycznych dla liczb zapisanych w BCD.

Powrót z przerwania maskowalnego powinien odbywać się poprzez RETI. Różnica pomiędzy obiema instrukcjami jest następująca, wskutek wykonania RET ze stosu pobierana jest wartość odpowiadająca adresowi powrotu, a następnie PC przyjmuje tą wartość (czyli następuje skok pod pobrany adres). W przypadku powrotu z przerwania zależy nam jednak nie tylko na odtworzeniu wartości PC, lecz także wszystkich pozostałych rejestrów — bowiem musimy odtworzyć całkowicie stan procesora sprzed wykonania przerwania. Stąd też przywracana jest nie tylko stara wartość PC, ale też wszystkich pozostałych rejestrów. Bez rozkazu RETI można się obyć, ale trzeba wówczas albo dbać o to, aby podczas obsługi przerwania nie naruszyć zawartości rejestrów, albo (i) własnoręcznie, przy pomocy innych instrukcji, zadbać o odtworzenie ich stanu. Uważam, że skoro mamy do dyspozycji instrukcję RETI, gra nie jest warta świeczki.

Efekty dźwiękowe znane nam dobrze z różnych gier programowane są „na piechotę” na podstawie dokładnej znajomości charakterystyki dźwięku, jaki ma być wytworzony. W przypadku mniejszych wymagań można efektów np. wybuchu poszukiwać „na wyczucie” poprzez eksperymentalny dobór pętli sterujących membraną głośnika — wszystko to oczywiście z poziomu asemblera! Pisałem już w tej rubryce, że w ZX Spectrum procesor bezpośrednio przekazuje wysokie lub niskie napięcie na port odpowiadający membranie zainstalowanego głośniczka. Stąd tyle kłopotów.

Przykładem uzyskania nietypowych efektów drugą z podanych metod może być program „Spectrum i klakson”, zamieszczony w nrze 2/86 naszego miesięcznika. Warto może z niego skorzystać.

Marcin Waligórski

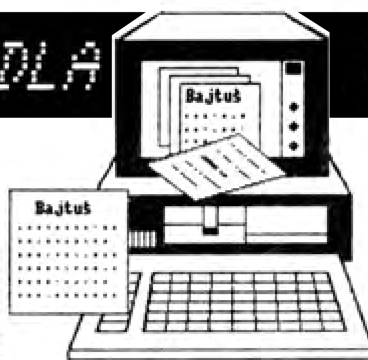


Posiadaczką głównej nagrody w Świątecznym Konkursie "Bajtka" została pani Bożena Potoczny.

— To był poprostu przejaw sprawiedliwości losu — powiedziała nam pani Bożena, mam 15 letniego Bartka i 13 letniego Szymona. Komputer wylosowany przez nas w Świątecznym Konkursie "Bajtka" jest dla moich chłopców jakoby rekompensatą za rozpacz, z jaką przyjęli niedawną kradzież ich Atari. Teraz znów mają swoją ulubioną zabawkę.

TYLKO DLA PRZEDSZKOLAKÓW

GAZETA



Cześć Maluchy!

Zauważyliście zapewne, że w tym numerze "Bajtka" bardzo wiele miejsca poświęcono programon do redagowania czasopism. My Przedszkolaki nie będziemy gorsi i zrobimy sobie własną gazetę.

Do wydawania gazety na papierze potrzebna jest drukarka. Nam wystarczy komputer, telewizor lub monitor i LOGO.

Powiedzmy, że nasza gazeta będzie się nazywała "Bajtuś". Trzeba więc zaprojektować jej winietę, czyli tytuł.

```
to winieta
pr [......................................]
pr [............... BAJTUS ...............]
pr [tygodnik Szafy z Zabawkami ...... nr 1]
pr [......................................]
end
```

Redagowanie pisma rozpoczniemy – zupełnie jak w prawdziwej gazecie – od napisania artykułów. Mogą one wyglądać na przykład tak:

```
to kronika.towarzyska
pr [........ KRONIKA TOWARZYSKA ........]
pr [Lalka Barbie urządza w sobotę bal na]
pr [drugiej półce. Zaproszenia otrzymali]
pr [między innymi Pajacyk i Miś. Na bal]
pr [nie została natomiast zaproszona lalka]
pr [Zuzia (ta nieduża) . Podobno lalki]
pr [pokłóciły sie ostatnio o nowy serwis]
pr [do kawy.]
end

to kronika.wypadków
pr [....... KRONIKA WYPADKOW .......]
pr [Chwile grozy przeżyli wczoraj w nocy]
pr [mieszkańcy najwyższej półki. Niesforna]
pr [Małpka Miki zamiast spać postanowiła]
pr [wybrać się na spacer i spadła z półki]
pr [na podłogę. Dopiero rano znalazły ją]
pr [dzieci. Teraz Miki leży w łóżeczku]
pr [i leczy zwichnięty ogon.]
end

to ze.świata
pr [.......... ZE SWIATA ..........]
pr [Lalki Krysi z II b dostały nowy domek.]
pr [Są w nim piękne mebelki: fotele,]
pr [łóżeczka. W kuchni wiszą piękne garnki]
pr [i wielka patelnia. Najładniejsze]
pr [jednak sa różowe firanki w kwiatki.]
end

to program.podwórkowy
pr [........ PROGRAM PODWORKOWY ........]
pr [6 00 - Pan Mleczarz przynosi mleko]
pr [8 00 - Karmienie Burka]
pr [9 00 - Przychodzi pan Listonosz]
pr [11 00 - Mama wiesza pranie]
pr [14 00 - Dzieci wracają ze szkoły]
pr [16 00 - Wyrzucanie smieci]
end

to wielki.wyścig
pr [......... WIELKI WYSCIG .........]
pr [Samochody z trzeciej półki odbyły]
pr [wielki wyścig Dookoła Dywanu. Wygrała]
pr [nieoczekiwanie Zielona Ciężarówka.]
pr [Wszyscy spodziewali się, że zwycięży]
pr [Wyścigówka z Czerwonym Pasem, ale na]
pr [ostatnim zakręcie odpadło jej koło.]
end

to mecz.piłkarski
pr [......... MECZ PIŁKARSKI .........]
pr [Wczoraj wieczorem Piłkarze na]
pr [Sprężynkach rozegrali mecz, który]
pr [zakończył się wynikiem 2 :2. Dogrywki]
pr [nie było, bo piłka potoczyła się]
pr [gdzieś pod szafę i dzieci nie mogły]
pr [jej znaleźć. Tata obiecał, że]
pr [przyniesie nową kulkę z łożyska.]
end

to kto.misiowi.urwał.ucho?
pr [..... KTO MISIOWI URWAŁ UCHO? .....]
pr [Wywiad z Misiem.]
pr [-Jak dzieci bawią się z Tobą Misiu?]
pr [-Niektóre są bardzo miłe. Spiewają mi]
pr [piosenki, szyją dla mnie ubranka,]
pr [kładą spać do łóżeczka.]
pr [-Powiedziałeś: Niektóre sa miłe.]
pr [-Bo są też takie, które bardzo brzydko]
pr [się bawią. Kiedyś taki jeden Krzyś]
pr [prawie urwał mi ucho! Mama musiała mi]
pr [go przyszywać.]
pr [-Czy to była Twoja najgorsza przygoda?]
pr [-Najgorzej było, gdy dzieci]
pr [postanowiły mnie wykąpać. Przez dwa]
pr [tygodnie wisiałem potem obok]
pr [kaloryfera. Już myślałem, że nigdy]
pr [nie będę miał sucho w brzuszku.]
pr [A potem to szczotkowanie futra... Brrr!]
pr [-Dziękuję za rozmowę Misiu.]
end

to nowa.buda.Burka
pr [....... NOWA BUDA BURKA .........]
pr [Burek dostał nową budę. Ładna?]
end

to zagadka
pr [............ ZAGADKA ............]
pr [Żołte, puszyste piórka,]
pr [okrągłe, grube brzuszki,]
pr [oczka jak dwa paciorki,]
pr [i płaskie, krzywe nóżki.]
end
```

W prawdziwej gazecie powinny znaleźć się ilustracje. Spróbujmy narysować jak wygląda nowa buda Burka.

```
to rys.buda
cs
ht pu lt 90 fd 100 lt 90 fd 50 lt 180
pd fd 120 rt 45 fd 100 rt 90 fd 100
rt 45 fd 120 rt 90 fd 140
pu home pd
repeat 12 [fd 17 lt 30]
end
```

Mamy już wszystkie materiały. Teraz wystarczy porozmieszczać je na poszczególnych stronach naszej gazety.

```
to strona1
ts
winieta
pr []
kronika.towarzyska
pr []
ze.świata
pr []
pr []
pr [..................................1]
end

to strona2
ts
kto.misiowi.urwał.ucho?
pr [] pr [] pr []
pr [..................................2]
end

to strona3
ts
kronika.wypadków
pr []
program.podwórkowy
pr []
zagadka
pr [..................................3]
end

to strona4
cs
rys.buda
nowa.buda.Burka
end

to strona5
ts
pr [...sport ... sport ... sport ...]
pr []
wielki.wyścig
pr [] pr []
mecz.piłkarski
pr [] pr [] pr []
pr [..................................5]
end
```

Od tej chwili odpowiedzią na polecenia: "strona1", "strona2"... będzie wyświetlenie na ekranie żądanej strony. Możemy jeszcze nauczyć komputer, aby na rozkaz "gazeta" wyświetlał pierwszą stronę.

```
to gazeta
strona1
end
```

Myślę, że nie zabraknie Wam fantazji w wynajdywaniu pomysłów do takich gazetek.

Romek

REWOLUCJA PRZY FLEET STREET

Dokończenie ze str. 32

a w ciągu ostatnich kilku lat stracił 75 tys. miejsc pracy. Np. w zecerni „Timesa” i „Sunday Timesa” pracowało przy Fleet Street ponad 600 ludzi, teraz w Wapping — już dla czterech tytułów — większą pracę wykonuje tylko 139 osób. Eddie Shah, właściciel pierwszego kolorowego w Anglii dziennika, robionego nie tylko za pomocą komputerów, ale lasera i satelity, zatrudniał 600 osób w tym 125 dziennikarzy. „Daily Telegraph”, o podobnym nakładzie — 1,2 mln egzemplarzy — zatrudniał wówczas, drukując gazetę tradycyjnie — 3,5 tys. ludzi.

Nowe technologie wprowadza się dlatego, żeby obniżyć koszty produkcji i zredukować zatrudnienie. Jak więc rozwikłać tę sprzeczność interesów?

Do rewolucji technologicznej, twierdzili związkowcy trzeba przygotować kraj, zorganizować ją tak, aby nie pogłębiała tragicznej sytuacji na rynku pracy, nie eliminowała z dnia na dzień całych zawodów i specjalności. W Anglii liczba bezrobotnych przekroczyła 4 miliony. Konserwatyści w walce ze związkami, a w rzeczywistości z Partią Pracy, nie liczą kosztów i konsekwencji społecznych. Co bowiem ma robić 45-50 letni człowiek, który nawet dostał kilkanaście tysięcy funtów odszkodowania i przez rok zasiłek dla bezrobotnych? Za wcześnie aby być rencistą, za późno, żeby uczyć się nowego zawodu. Dla drukarzy zresztą, zmiana zawodu zawsze idzie w parze z degradacją. Dzięki przywilejom wywalczonym jeszcze wówczas, gdy wśród niewykształconej klasy robotniczej stanowili wysoko wykwalifikowaną elitę zarabiają dobrze: 250 do 350 funtów tygodniowo, jeśli dodatkowo pracują w soboty (średnia angielska ok. 200 funtów tygodniowo).

Kilkumiesięczny protest drukarzy poparł także Związek Dziennikarzy (NUJ). Solidarność z kolegami, mówili przywódcy związku, jest konieczna jak nigdy. Nie chodziło o ratowanie Fleet Street, nostalgiczne westchnienia za atmosferą jej okolicy, pubami, starymi murami; prasie groził koniec niezależnego dziennikarstwa. Do czasu bowiem pojawienia się magnatów prasowych — 16 spośród 17 tytułów o zasięgu ogólnokrajowym jest w rękach siedmiu ludzi — właściciele kochali to, co robili. Jeśli konkurowali ze sobą, to korzystał na tym czytelnik. Dzisiaj, czego nie ukrywają właściciele, gazeta jest produktem, o wszystkim decyduje pieniądz. Mniejszą wagę przykłada się więc do jakości tekstów tak, że przyjęło się powiedzenie: „Nieważna jakość, upewnij się, czy długość się zgadza”.

Po dwóch latach od wydarzeń w Wapping większość tytułów centralnych drukuje się w Anglii wykorzystując nowe technologie składu i druku. Kolejne kilkadziesiąt tysięcy ludzi straciło pracę, ale redukcje przeprowadzono już mniej burzliwie; wypłacano większe odszkodowania, organizowano kursy, m.in. komputerowe, stwarzając możliwości przekwalifikowania się, powstało wiele małych firm drukarskich, gdzie część zwalnianych znalazła zatrudnienie.

Czy nam, w Polsce grozi podobna rewolucja technologiczna? Na razie nie, bo po pierwsze... nie mamy armat. Prawa ekonomiczne są jednak nieubłagane i prędzej czy później, któryś z naszych zakładów poligraficznych stanie przed koniecznością zredukowania sześciu siódmych personelu. Można jednak przewidywać — biorąc pod uwagę dotychczasową praktykę gospodarczą — że trzech czwartych załogi nie będzie można zwolnić; jedynych żywicieli rodzin, zasłużonych, odznaczonych, tych przed emeryturą, itp. Nawet zwalnianym nie będzie też grozić, jak w Anglii, bezrobocie. Wszak ciągle żyjemy w kraju, gdzie jest więcej miejsc niż chętnych do pracy.

Wojciech Markiewicz

REWOLUCJA

PRZY FLEET STREET

W piątek, 24 stycznia 1986 r. Rupert Murdoch, właściciel czterech czołowych angielskich tytułów prasowych — „The Times", „The Sunday Times", „The Sun" i gazety niedzielnej "News of the World" — zwolnił z pracy sześć z siedmiu tysięcy drukarzy, techników i urzędników, których dotąd zatrudniał przy redagowaniu i druku tych pism. Jak doszło do tego, że z dnia na dzień sześć siódmych personelu straciło pracę a gazety ukazywały się w niezmienionej objętości i nakładzie? Uproszczeniem byłoby napisać, że 6 tys. ludzi straciło pracę za sprawą komputerów — nowych technologii składu i druku.

Premier Margaret Thatcher całkowicie poparła Murdocha. Stwierdziła m.in., że wydawcy mają całkowite prawo korzystania ze wszystkich instrumentów prawnych, jakie są do ich dyspozycji. Pani Thatcher dodała, że zbyt długi już był opór wobec wprowadzania zmian w technologii druku.

Tymczasem Centrala Związków Zawodowych (TUC) podjęła procedurę zawieszenia Związku Elektryków (EETPU) za to, że jego członkowie podjęli się drukowania gazet Murdocha, zastępując strajkujących drukarzy. Z kolei przywódca elektryków ostrzegł, że wystąpi na drogę sądową przeciwko TUC, jeśli jego związek, który walczy o miejsce pracy dla swoich członków, zostanie zawieszony.

Stworzyło to zupełnie nową sytuację w tradycyjnych konfliktach pracodawca-związki. Tym razem już nie sam magnat prasowy kontra drukarzom, ale magnat ze związkiem elektryków przeciwko drukarzom. Gdyby TUC wydaliła EETPU ze swoich szeregów, umocniłoby to dążenia odśrodkowe innych związków i tendencje do utworzenia konkurencyjnej centrali związkowej. Byłoby to kolejnym, po niepowodzeniu strajku górników, osłabieniem ruchu związkowego, a w konsekwencji osłabieniem opozycyjnej Partii Pracy.

Przy Fleet Street, słynnej londyńskiej ulicy, w okolicach której wydawano większość brytyjskich dzienników i tygodników o zasięgu ogólnokrajowym, jeszcze w połowie lat osiemdziesiątych drukowano gazety tradycyjnie. Mimo, że w Europie Zachodniej, nie mówiąc o Stanach Zjednoczonych, nowe techniki składu i druku wprowadzono już na początku lat siedemdziesiątych, tutaj w niejednej drukarni składano teksty na linotypach wyprodukowanych jeszcze w ubiegłym stuleciu..

Właściciele wielu tytułów brytyjskiej prasy bali się inwestować, bali się ciągle jeszcze silnych związków zawodowych, strajków. Opór związków wobec wprowadzania zmian w technologii składu i druku nie był irracjonalny. Christopher Robbins, sekretarz londyńskiego oddziału SOGAT powiedział mi w marcu 1986 r.: — To jest problem społeczny, a nie technologiczny. Technologie są wspaniałą rzeczą, pod warunkiem, że zyski z ich zastosowania nie będą wpadały tylko do kieszeni właściciela. Jesteśmy za nowymi technikami, ale musimy utrzymać jak największą liczbę miejsc pracy.

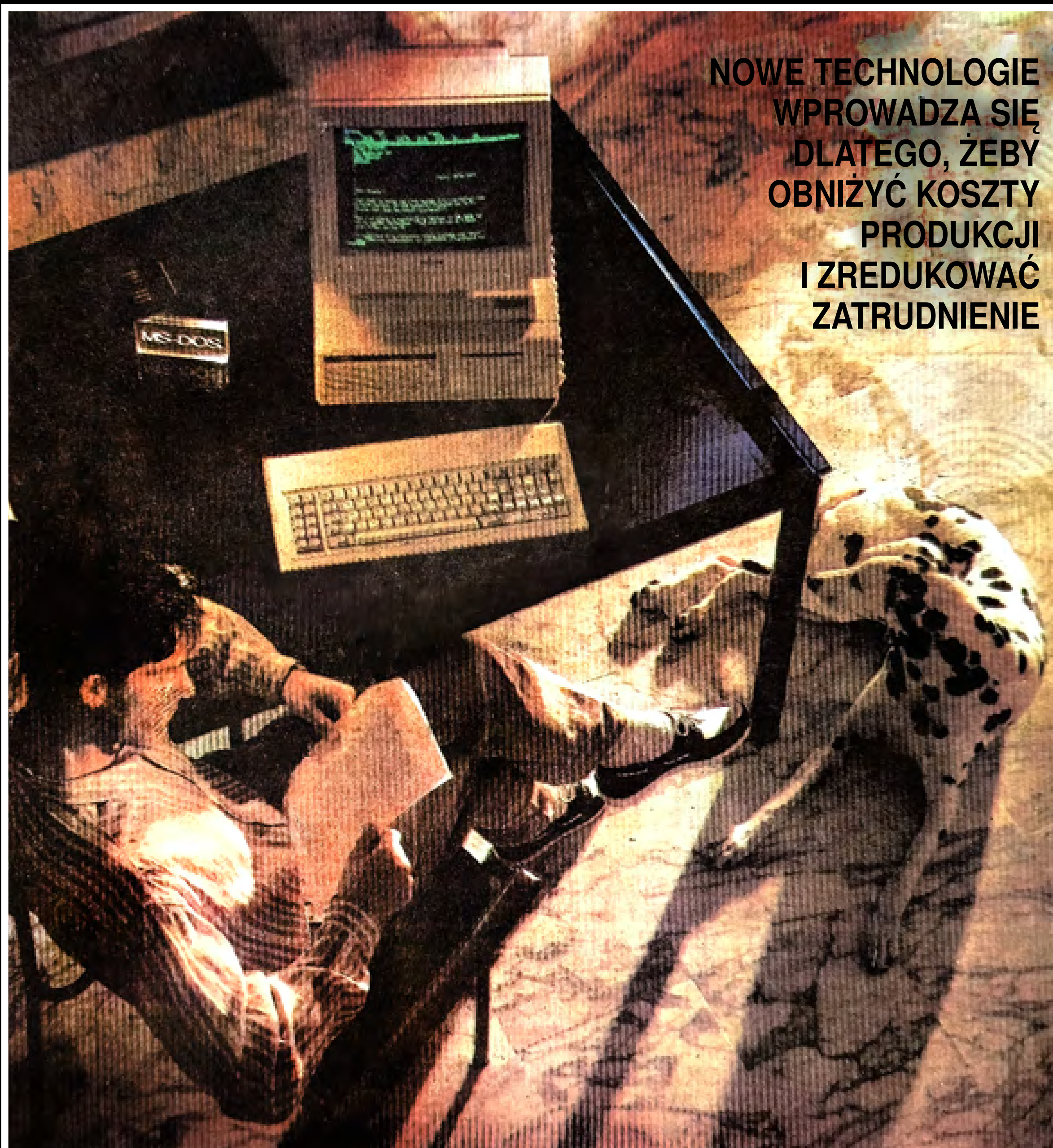
Były to czasy, kiedy w samym tylko Londynie było 4000 tys. bezrobotnych,

Dokończenie na str. 31

Od 1980 r. Rupert Murdoch (57 lat), którego imperium prasowo-telewizyjne działa na trzech kontynentach (Australia, Ameryka, Europa — 80 gazet i czasopism, 7 stacji TV w USA, kontrola nad amerykańską wytwórnią filmową i telewizyjną 20-th Century Fox) zaczął budować, kosztem 100 mln funtów szterlingów, drukarnię w Wapping, londyńskiej dzielnicy doków. Zapewniał wówczas swoich pracowników, że będzie tutaj drukowana nowa popołudniówka. Potajemnie sprowadzono z USA, z komputerowej firmy ATEX, w nieoznakowanych skrzyniach urządzenia do nowej drukarni. Jednocześnie Murdoch, który zdecydował, że Wapping będzie funkcjonowała bez drukarzy, zastanawiał się wraz ze swoimi doradcami prawnymi, jak najtańszym kosztem pozbyć się 6 tys. ludzi.

Doradcy prawni stwierdzili, że najtaniej będzie wymówić im pracę podczas strajku. Sformułowano więc projekt nowego kontraktu tak, żeby był on nie do przyjęcia (pracodawca nie uznaje związków, strajków, ma nieograniczone prawo zawieszania, karania i zwalniania z pracy, itp.). Projekt ten, stawiający związki SOGAT i NGA bardziej w roli obrońcy pracodawcy niż swoich członków, zgodnie z oczekiwaniami doradców został odrzucony i drukarze rozpoczęli strajk. Wówczas to R. Murdoch zwolnił 6 tys. pracowników, a redakcje i drukarnie swoich czterech tytułów przeniósł do Wapping. gdzie pracowali już członkowie związku elektryków (EETTPU). Była to batalia o 50 mln funtów oszczędności rocznie, na samych tylko wynagrodzeniach dla drukarzy.

SOGAT i NGA zaczęły więc blokować dystrybucję czterech tytułów drukowanych w Wapping. Ale Murdoch szybko złożył przeciwko obydwu związkom pozew. Sąd, trzymający się litery przeforsowanego przed kilku laty przez konserwatystów prawa, orzekł sekwestrację majątku związku SOGAT, zamroził jego konto bankowe oraz skazał na grzywnę w wysokości 25 tys. funtów. Okazało się bowiem, że wszelkie akcje przed drukarnią w Wapping są nielegalne, gdyż w świetle prawa można pikietować wyłącznie swój zakład pracy, czyli opuszczone drukarnie i redakcje w okolicach Fleet Street.



NOWE TECHNOLOGIE WPROWADZA SIĘ DLATEGO, ŻEBY OBNIŻYĆ KOSZTY PRODUKCJI I ZREDUKOWAĆ ZATRUDNIENIE



www.t2e.pl

Seria „Reduks" ma na celu przetworzenie – pełne zdigitalizowanie prasy komputerowej z początków okresu informatyzacji naszego kraju.

W odróżnieniu od plików w formacie djvu i cbr gdzie format papierowy jest tylko zeskanowany i połączony w jeden dokument, **reduks polega na ponownym przełamaniu dokumentu, wraz z ponowną obróbką graficzną - często z ich ponowną kreacją**. Tam gdzie to możliwe zachowana jest grafika z pierwowzoru, jednak jeśli materiał jest zbyt złej jakości - zostaje on podmieniony (oczywiście jeśli jest dostępny) na równoważny. Jeżeli uda się odnaleźć błąd w zamieszczonych listingach są one poprawiane tak, aby mogły zostać uruchomione na emulatorach lub mikrokomputerach.

Rozpocząłem ten cykl od „Bajtków", jednak mam nadzieję, że uda mi się wzbogacić kolekcję zdigitalizowanych i ponownie przełamanych periodyków także o inne ważne tytuły z lat ubiegłych, o których powoli zapominamy, a bez których współczesny obraz komputeryzacji wyglądałby zgoła inaczej.

Wydania „Reduksowe" są bezpłatne, nie roszczę sobie żadnych praw do zawartości przetwarzanych magazynów, jednak ilość czasu i pracy jaką wkładam aby mogły one trafić w Twoje ręce jest na tyle spora, że pozwolę sobie napomknąć iż możesz wesprzeć mnie przekazując dotację w dowolnej wysokości.

Jeśli posiadasz materiały fotograficzne lub edytorskie, które są lepszej jakości niż prezentowane w numerach, daj znać.



W ramach wolnego czasu odtwarzam stare magazyny komputerowe i prowadzę serwis emulacyjny try2emu. Jeżeli uważasz, że moja publiczna działalność możesz postawić mi kawę, a kawę lubię;)



https://buycoffee.to/t2e

Osoby, które wspierają serwis datkami* mają dostęp do reduksów w wysokiej rozdzielczości z usuniętymi reklamami.

W zależności od wysokości datku każdy otrzymuje czasowy dostęp do reduksów w pełnej rozdzielczości na miesiąc, 3 miesiące lub 6 miesięcy.
Osoby, które aktywnie wspierają projekt mają dostęp nieograniczony.